M_PLAYER V4.25

Documentation

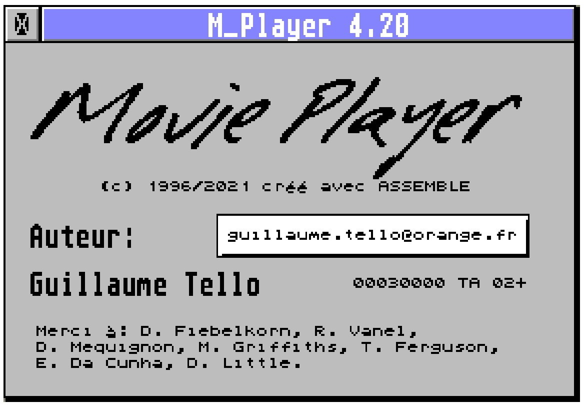
Video Player, Maker and Disassembler
For the Atari 68030+ computers

Author : guillaume.tello@orange.fr 1996-2021
WEB Pages:
- (in english) https://gtello.pagesperso-orange.fr/mplayere.htm
- (in french) https://gtello.pagesperso-orange.fr/mplayerf.htm

Thanks to **Dieter Fiebelkorn** for sharing its original C work on AVI and MOV.
Thanks to **Didier Mequignon** for sharing IV32 and IMA4 codecs.
Thanks to **Douglas Little** for providing the fast Byte to Planes conversion for TT-Low resolution
Thanks to **Rémi Vanel** for FLH and partnership on his Mountain software using M_PLAYER.
Thanks to **Olivier Landemarre** for helping me on multitasking and GEM management.
Thanks to **Martin Griffiths** for his fast DCT routine on MPEG.
Thanks to **T. Ferguson** for his web page with the idea of the CRAM encoder.
Thanks to **Nima Montaser** for his partnership on his great Infopedia driver using M_PLAYER.
Thanks to **Eric Da Cunha** for his module JPEG_68K.RIM (from parx M&E system).

Contents

I Introduction


M_Player 4.20

Movie Player

(c) 1996/2021 créé avec ASSEMBLE

Auteur: guillaume.tello@orange.fr

Guillaume Tello 00030000 TA 02+

Merci à: D. Fiebelkorn, R. Vanel,
D. Mequignon, M. Griffiths, T. Ferguson,
E. Da Cunha, D. Little.

M_PLAYER is a **video player for Atari** computers running TOS or similar operating system on a Motorola **68030 CPU or better**.
It is also a **video disassembler and encoder** that handle various types of formats.

It was written mostly for the Atari TT with a graphic card in 16 bits colors.
The goal of the program was not to make a GEM-friendly application. Knowing that a 32MHz machine is limited in power, I wanted to take the best of the hardware to get a good replay experience : not jerky sound, not jerky display.
That’s why M_PLAYER doesn’t use the VDI for graphic output when direct access is available, reserves the screen, seldom shares the CPU under a multitasking environment.

M_PLAYER can be used as a **GEM program** but also as a **desk accessory**. Just rename it as M_PLAYER.ACC and copy it in the root of your boot drive. The file M_PLAYER.RSC must be located in the same folder.

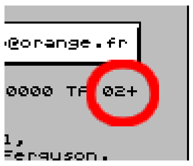
I.1 RSC discussion

The program comes with **different ressources files** according to **different languages**. They are named M_PLAYER.nnn where nnn is the country code:

000	English (US & UK)
001	German (and Swiss German)
002	French (ans Swiss French)
004	Spanish
005	Italian

The program **determines the country code** and tries to load the corresponding ressource. If this one is not available, then it tries to load a default M_PLAYER.RSC. Again, if this one is not available, then you get an alert and the program stops. For now, two ressources are available:
M_PLAYER.RSC, the default english one
M_PLAYER.002, the french one.

If you don't know the **country code of your system**, it is displayed in the **credits dialog** box at the end of the line using the small font:


@orange.fr
0000 TA 02+
l,
Ferguson.

Here **"02+"** means french system and that M_PLAYER.002 could be loaded.
For example, **"05-"** would mean italian system but M_PLAYER.005 was not found and the default M_PLAYER.RSC was loaded.

I.2 Graphic requirements

- ✔ An **Apollo Vampire Standelone V4+** in 16 bits (the best) but other mdoes are supported.

- ✔ An **Atari TT with a graphic card in 16 bits** is the best.
- ✔ A **Falcon (CT060 supported, CTPCI+Radeon too) in High Color** mode.

In this mode, the video is replayed in a window if it fits the screen, else the whole screen is used.

- ✔ An **Atari TT with a graphic card in 256 colors**.

In this mode, the screen turns black, the video is centered and any animation with more than 256 colors is replayed in grey levels.

- ✔ An **Atari TT with no graphic card.**

The planes management of the Atari video memory makes this mode slower than the others. Anyway, the TT switches to TT Low to replay animations in 256 colors/greys. For earlier animations on Atari, the corresponding video mode is used (ST Low for example).

- ✔ An **Atari TT with the AlberTT** graphic card.

As the card is limited to 16 colors, the displayed is redirected to the standard TT output and works with the TT Low resolution. So you need a second monitor or a switch.

- ✔ A **Falcon in any other mode than High Color.**

In this mode, the Falcon switches to 320x240xTrue Color hoping that the animation fits the screen. This is included but not certified to work...

- ✔ Other **machines with 256/TC modes** (Hadès, MagicMac)

They work as the TT.

- ✔ Any **machine with 24 or 32 bits display.** (Aranym for example)

In this mode, M_Player has no specific routines, so the 16 bits native routines are used, and, after a reencoding, the VDI is used to display every image.

I.3 Sound requirements

Sound is replayed via the **DMA system** on Atari TT, Falcon and Aranym.

If no DMA sound is available, then the **Yamaha** chip is used (This is really slow with a low quality). If a **Psound card** (parallel sound card) is detected with the PSND cookie, then it is used !

See at Set Options and Paths.

On the Apollo Vampire, the **SAGA sound chip** is fully supported and works like the DMA on the TT.

I.4 OS requirements

M_Player was primarly intented to run with the **standard TOS** system.

It supports **MagiC, MagicMac, Mint, Geneva and EmuTOS**. But in any mutlitasking system, the program reserves the ressources and don’t allow the user to work on another application while it is replaying a video.

However, while the dialog boxes are displayed, or during a pause, or when an animation is encoded, then it can be considered as a multitasking application.

There is no more problem with multitasking systems and the Control key as from version 4.00 M_PLAYER doesn’t use the supervisor mode. All routines have been rewritten to play in user mode.

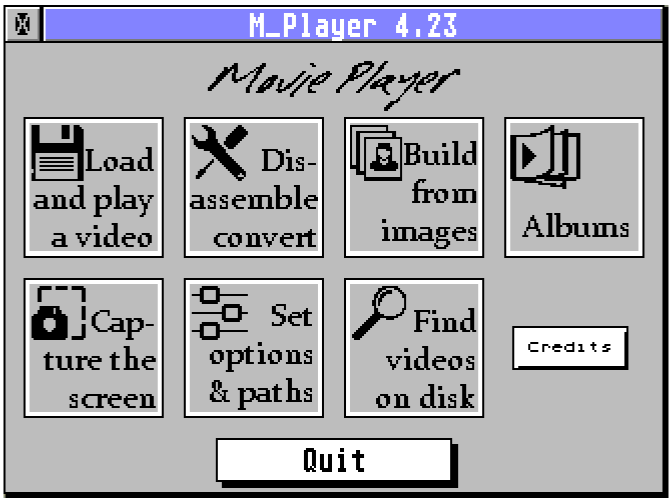
I.5 M_PLAYER as an accessory

On some systems, the player can't find its ressource file or option file when installed as an accessory.

So, you have to respect this organization:

- ➔ M_PLAYER.ACC in the root of the boot disk or in the ACC folder (under Mint for example)
- ➔ M_PLAYER.RSC, M_PLAYER.OPT, M_PLAYER.AVR, JPEG_68K.RIM in the root of the boot disk, even if the executable is located in the ACC folder.

II General Menu



When you run M_PLAYER, this menu appears giving access to every feature of the pogram.

You can:

- **Load and play a video**, see the list of supported formats. The program will do its best according to the capabilities of your system concerning video and sound.
- **Disassemble or Convert**, you can turn a video into a set of individual images, a sound file or just convert the video from one format to a limited target of formats.
- **Build a video from images**, you can reassemble a set of images and eventually a sound into various formats of videos.
- **Albums**, let you create or open an album with a set of videos that you have organized and can play without searching on the disk.
- **Capture the screen to an animation**, the program lets you define a rectangle and capture the screen (manually or automatically) that is assembled on the fly into a video.
- **Set Options and Paths**, this is a very important part that you should rapidely fill with your own preferences to greatly ease your work
- **Search a disk for animations**, a search tool that returns a text file with every video found (selectable extensions) from a particular folder on disk.
- **Credits**, opens the credits panel.
- **Quit**, returns to the desktop.

It's time to have a look at every entry of this menu...

III Load and play a video

This **opens the fileselector** where you can select a video to be played.

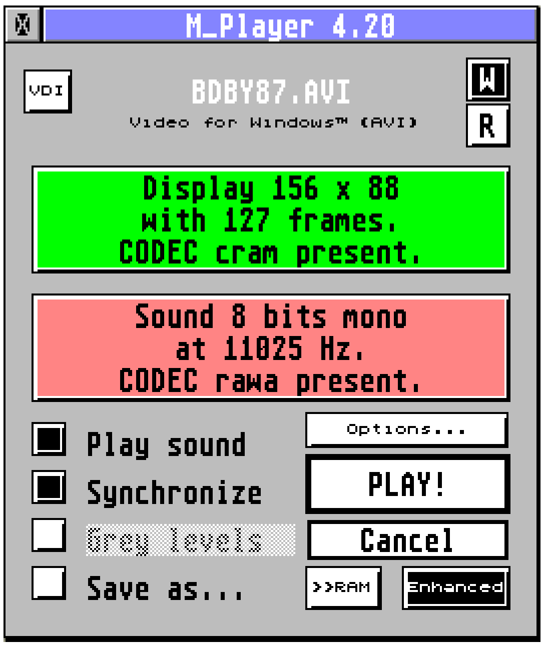
When the program reaches the **end of a video**, it **returns to the fileselector**, so you can chain rapidely animations without going back to the menu.

To go **back to the menu**, just **clic on <u>Cancel</u>** in the fileselector.

III.1 AVI, MOV and Extended FLM

Those three types have a similar behaviour : they come with images and sound.

After the file has been selected, the main dialog opens and displays informations about the video. It also tells you wether the display and sound codecs are supported or not.



BDBY87.AVI
Video for Windows™ (AVI)

Display 156 x 88
with 127 frames.
CODEC cram present.

Sound 8 bits mono
at 11025 Hz.
CODEC rawa present.

<u>Several buttons are available :</u>

VDI button

If set, you're in 24 or 32 bits display and the VDI will be used instead of direct routines (or in 16 bits under emulators such as Aranym). *This button is read only.*

W button

If set, the display can be done within a window. You can deselect it to get a black background.

R button

If set the animation will be repeated until the Control key is pressed.

>>Ram button

If set, the whole video is loaded first in memory, it speeds up the display. If not enough memory available, then this button is disabled.

Play Sound button

If set, the sound will be played, else mute.

Enhanced / Record button

See chapter Enhanced files.

Synchronize button

If set, the images will be synchronized with sound or with the file time settings. If not set, images are replayed at full speed.

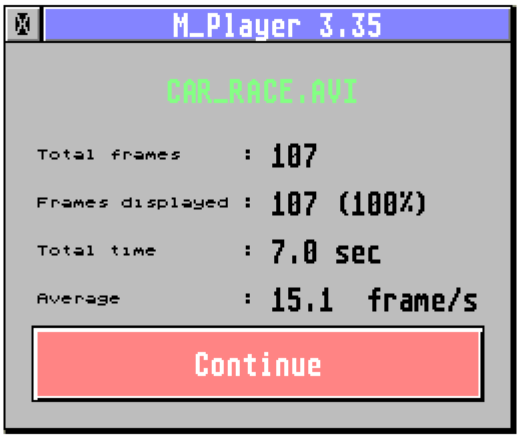
Save As button

Opens a dialog where you can disassemble or convert a file to another format. This can be done directly from the General Menu entry: **Disassemble or Convert.**

Options button

Opens the option panel, see at Set Options and Paths.

Play ! Button



```
M_Player 3.35

CAR_RACE.AVI

Total frames       : 107
Frames displayed   : 107 (100%)
Total time         : 7.0 sec
Average            : 15.1  frame/s
```

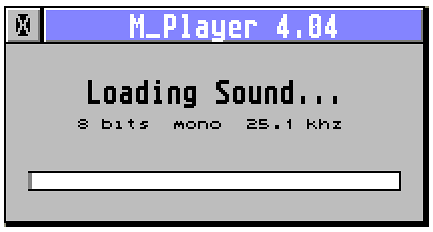
Replays the animation with your settings.

At the end of the replay, you get some informations about the quality.

How many frames per second were displayed.

How many frames were actually displayed compared to the total. For sync needs, some frames may be skipped.

Hold CONTROL to stop animation prematurely. Hold Alt+Control to Pause. See Pausing.

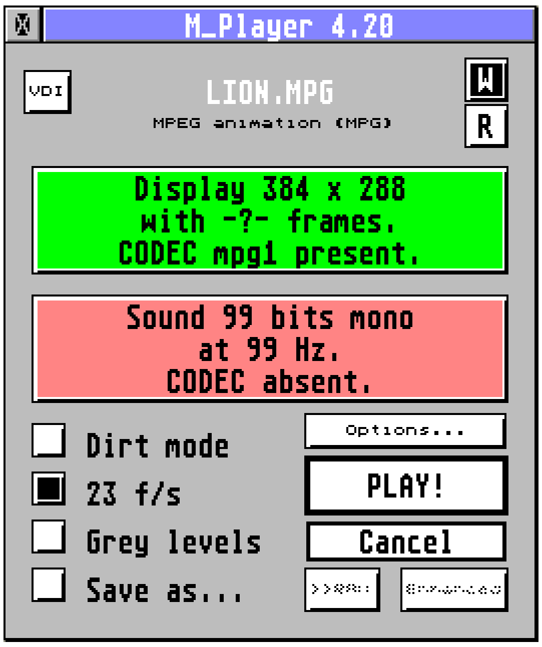


```
M_Player 4.04

Loading Sound...
8 bits  mono  25.1 khz
```

If a sound is to be loaded, a little dialog **"Loading sound..."** appears. There you can read the hardware informations for the sound. Typically, if you use the DMA, the frequency will be 12,5 or 25 or 50KHz. If you use the Yamaha or PSound card, it will be 9,6 KHz.

If you don't have the time to read the informations, you can **hold the SHIFT key** while clicking on **Play !**, the dialog will remain until you release the key.

III.2 MPEG files



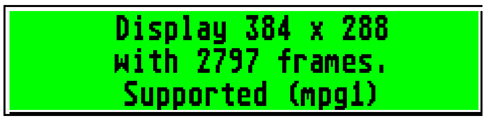
```
M_Player 4.20

LION.MPG
MPEG animation (MPG)

Display 384 x 288
with -?- frames.
CODEC mpg1 present.

Sound 99 bits mono
at 99 Hz.
CODEC absent.
```

When loading an MPG file, the **number of frames is not available**. This is the default.

To get it, M_Player has to parse the whole file and that can **take a lot of time**.

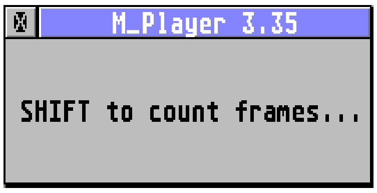
But if you really want this information, when the fileselector is openend, clic on you MPG file and then **hold the Left Shift key while validating** your choice.

At this moment, you'll see M_Player working for a while and finally display the complete informations :

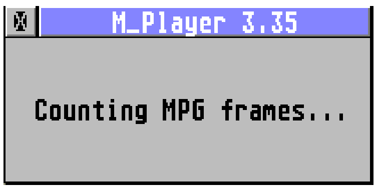


```
Display 384 x 288
with 2797 frames.
Supported (mpg1)
```

According to the power of your machine, M_PLAYER decides weather or not it will count the frames. If the animation is considered small enough to require only a few seconds, frames will be counted. Else, you'll get this box during one second :


M_Player 3.35

SHIFT to count frames...


M_Player 3.35

Counting MPG frames...

If you hold Shift, then counting will be forced and during this operation, another box appears.

Take care ! For example, on a TT, one MB of animation will take approximately 1 second. So one minute for a 60MB video.

See at Set Options and Paths to force count.

Because **MPG replay is very slow** on a standard Atari machine, you can select :

Grey Levels button

If you select this, the MPG will only decode grey levels and ignore color data, this can speed up a bit the replay.

Dirt mode button

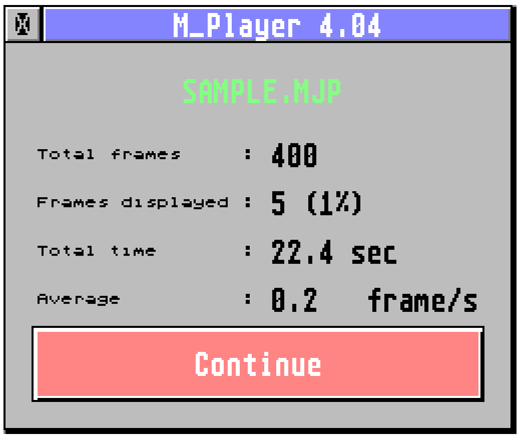
If set, then only a short part of the encoding is used to offer a reduced view of the animation in very low quality, but then again, more speed.

As an example, here are the times in seconds for the different modes for the above animation on a TT with graphic card:

FRANCE2.MPG	Color	Grey
Normal	173 sec	119,6 sec
Dirt mode	65,4 sec	46,6 sec

As you can see, **the sound is never supported in MPG files**.

III.3 MJPEG files/Codec in AVI-MOV


M_Player 4.04

SAMPLE.MJP

```
Total frames      : 400
Frames displayed  : 5 (1%)
Total time        : 22.4 sec
Average           : 0.2   frame/s
```

MJPEG (motion JPEG) frames can appear as:

- a **CODEC into AVI or MOV** files
- **standalone files**

and they all require the module **JPEG_68K.RIM** *(written by Eric Da Cunha)*.

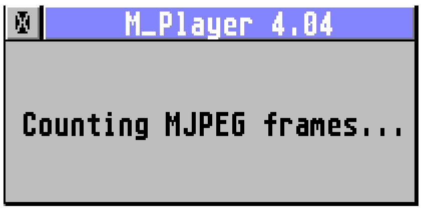
In the first case, nothing new, you can replay your AVI or MOV file as before. You'll just notice that on a real Atari, the frame decoding is slow, slower than the MPEG, those frames are time and power consuming.

As an example, you can see the stats of a MJPEG file 640×360 on a TT that should be replayed at 20 fps.

In the second case, you have **one or more concatenated JPEG images**, typically with extension JPG for single images and MJP (short for mjpeg) for movies. They often come from cameras as there is no need to build a specific file structure, JPEG files are just written one after the other.

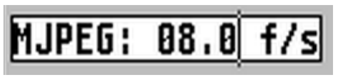
Two problems come there:

- the **MJPEG files don't have any timing** informations
- they **don't have the number** of frames information


M_Player 4.04

Counting MJPEG frames...

This is why you'll see the dialog **"Counting MJPEG frames..."** when you load one. This can take a while depending on the length of th file.


MJPEG: 08.0 f/s

And this is why the **option panel** lets you specify a default frame rate. See at Set Options and Paths.

This value runs from 0,1 to 99,9.

<u>Examples:</u>

- *With 0,1 fps, every frame will last 10 seconds, a slideshow.*
- *With 10 fps, every frame will last 0,1 second.*
- *With 99,9 fps, every frame will last 0,01 second.*

<u>Note:</u> *If you want to see **every image** with no synchro, then select **"Max Speed"** in the main dialog.*

<u>Note:</u> if a **JPEG** file contains **one single image**, then it remains displayed on the screen until you press **Control** or one **mouse button**.

III.4 GIF files

As the MPG, you don’t get the number of frames displayed unless the file contains the M_PLAYER extension. This happens only if the file was created by M_Player.

Two other extensions are supported :
DELAY : that sets the time between two frames
NETSCAPE : for a loop displayed

If a GIF has a loop extension, you can stop the display at the end of a loop using the Shift key.

III.5 Old Atari/PC animations files

- <u>Some are mute</u>

FLI, FLC, FLH

from Autodesk Animator (16, 256 or TC modes)

SEQ

from cyberpaint (ST Low)

PI1+DLT

from Cyber, select the PI1 file, the DLT must be in the same folder with same name.

(ST Low)

FLM

From Kinetic Microsystems (ST Low)

DL

from the PC world. Support for DL1 and DL2 files is provided.

- <u>Some have sound</u>

FLM

from Lexicor. (ST Low, ST Med, ST High)

This format has been extended by M_PLAYER and MP_STE with sound. That’s what I call Extended FLM. Both M_PLAYER and MP_STE can generate such files. This new format is adapted to ST Low and ST High video modes.

Sound has the DMA requirements :

- 8 bits mono/stereo
- frequencies 12,5kHz, 25kHz or 50kHz.

FLM

From Video Master, the digitizer for ST/Falcon. (160×100 in 16 or 256 colors)

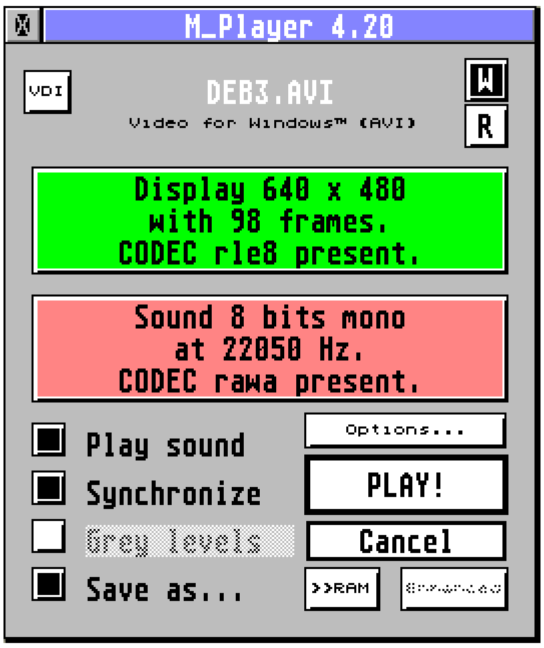
Sound is always 8 bits mono with the above frequencies.

CDH, CDL, CDV

From the CD of the Stratos french magazine. Animations in mono, 16 colors or 256 colors to suit ST High, ST Low or Falcon 256 colors resolution. For the last, a TIFF image with the same name in the same folder is required for the palette.

Sound is 8 bits mono in 12,5kHz.

IV Disassemble or Convert


M_Player 4.20

VDI

DEB3.AVI

Video for Windows™ (AVI)

W

R

Display 640 x 480
with 98 frames.
CODEC rle8 present.

Sound 8 bits mono
at 22050 Hz.
CODEC rawa present.

- [x] Play sound
- [x] Synchronize
- [ ] Grey levels
- [x] Save as...

Options...

PLAY!

Cancel

>>RAM

When you clic on the button from the General Menu, this opens the fileselector and then you're driven to the panel shown on the right with the file you want to work on. Select **<u>Convert!</u>**.

<u>Note:</u> *there was another way to do it before the General Menu was created. It is still supported:* manually set the button "**<u>Save As...</u>**" when you are replaying videos. Then you select **<u>Play!</u>**

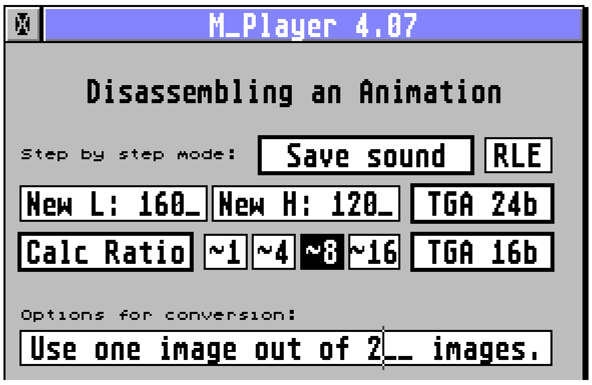
Here, I have loaded an animation with sound and 98 frames 640×480 pixels.

Let’s work on it !

IV.1 Disassembling a video

A video is a set of **images** and eventually a **sound**. All of this can be **taken apart** and saved separately.

Lets have a look at the top half of the dialog :


M_Player 4.07

Disassembling an Animation

Step by step mode: | Save sound | RLE

New L: 160_ | New H: 120_ | TGA 24b

Calc Ratio | ~1 | ~4 | ~8 | ~16 | TGA 16b

Options for conversion:

Use one image out of 2__ images.

Save Sound button

if available, you can save the whole sound as an **AVR file**. *(AVR is a type of sound file for Atari used by the ST Replay cards).*

TGA 24b and TGA 16b buttons

Those lead to the interactive display/save menu for the images. They will be saved as TGA files in 24 bits or 16 bits and eventually with the **RLE compression**.

<u>Note :</u> *RLE compression is based on the early specifications from TARGA. At the time, compressed blocks could bypass a line. In the late specifications, this is forbidden. Some image viewers will show a bad image with RLE compression. In case of doubt, disable it !*

If you want to modify the size of the images saved, you can fill **New L** and **New H** with your own values, else leave those fields blank. To keep the original ratio, just use :

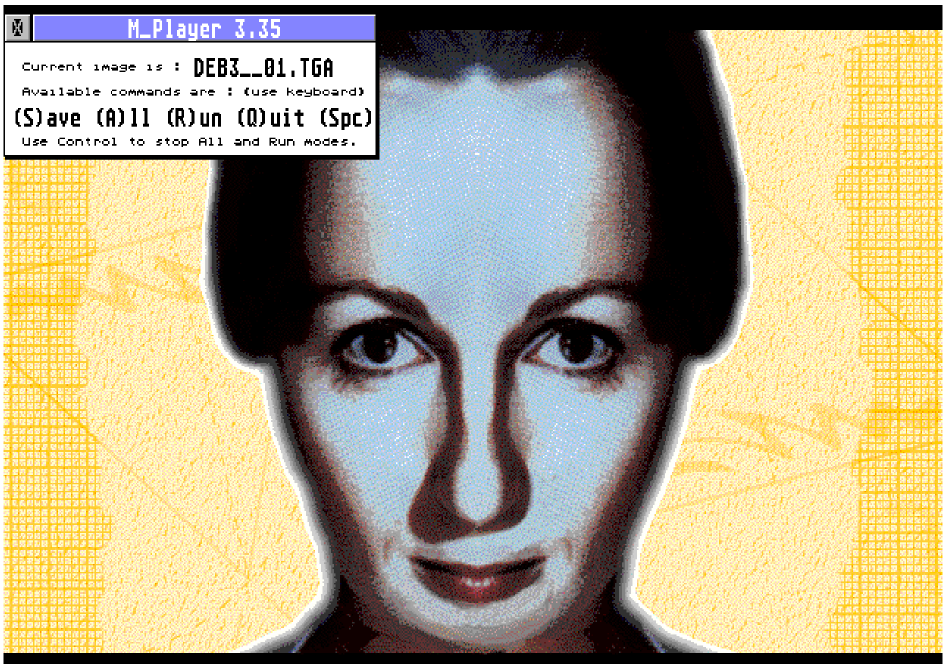
Calc Ratio button

Fill **New L** or **New H**, leaving the other empty. **Calc Ratio** will compute the blank value rouding the results to a 1, 4, 8 or 16 pixels boudary.

Interactive saving

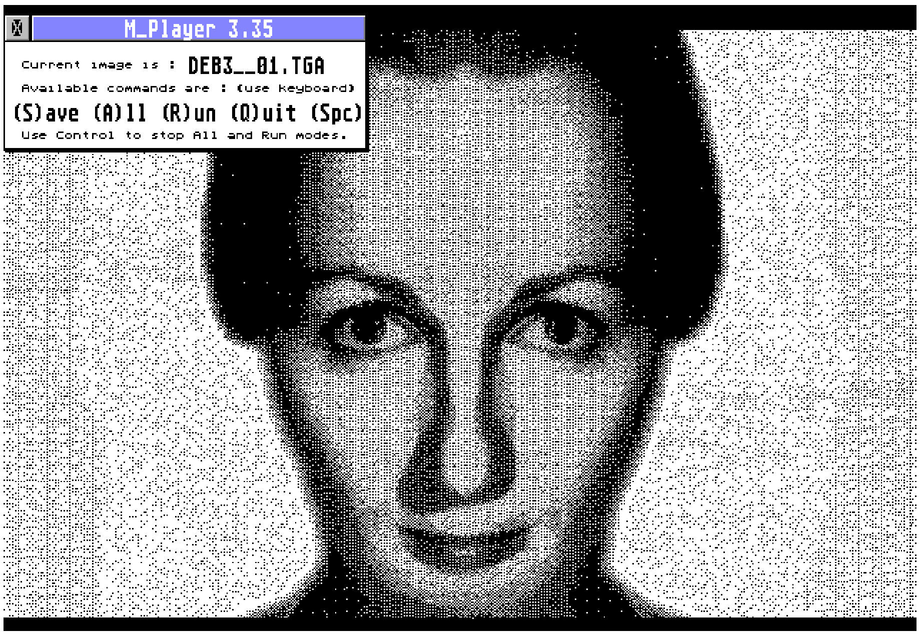
With **TGA 24b** or **TGA 16b** button, unless you have filled the auto settings with *Set Options and Paths*, you're driven to a fileselector. You have to give a generic name to your images and select the folder to save them. If you leave the filename blank, then the default **FRM00001.TGA** will be used.

If you want to change this name, just type the first letters and the digits and TGA extension will be automatically appended. For example, if you type VIDEO, then VIDEO001.TGA will be used as the first file name. Validate and you enter this screen :



```
M_Player 3.35
Current image is : DEB3__01.TGA
Available commands are : (use keyboard)
(S)ave (A)ll (R)un (Q)uit (Spc)
Use Control to stop All and Run modes.
```

For TGA 16/24 bits saving, the image must be decoded in those high color formats. If you're running on a **system with a maximum of 256 colors**, then the images are decoded internally with high color and just diplayed with a **monochrome filter** so you can still use the feature and know what the current image is.



```
M_Player 3.35
Current image is : DEB3__01.TGA
Available commands are : (use keyboard)
(S)ave (A)ll (R)un (Q)uit (Spc)
Use Control to stop All and Run modes.
```

Save

Save current image and go to the next according to the value you've entered in

Options for conversion:

Use one image out of 2__ images.

Let's call this value the STEP.

All

Save all images in sequence STEP by STEP without waiting for a key.
Hold **Control** to return to interactive mode.

Quit

Leave this screen and return to the General Menu (or the Fileselector if you entered manually this mode).

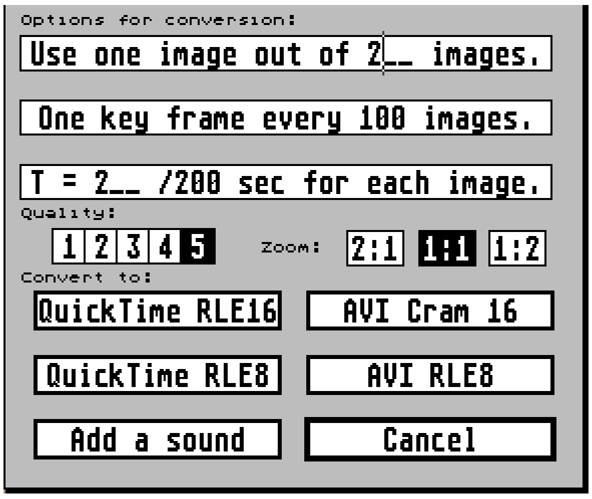
Run

Display images in sequence without saving.
Hold **Control** to return to step by step mode.

Space

Go to next image.

IV.2 Converting a video


Options for conversion:
Use one image out of 2__ images.
One key frame every 100 images.
T = 2__ /200 sec for each image.
Quality: 1 2 3 4 [✓ 5]
Zoom: 2:1 [✓ 1:1] 1:2
Convert to:

M_Player allows you to turn a video from one format to another **on the fly**, without saving tons of images.

This feature has been rewritten to be accepted by multitasking environments, the memory is correctly shared.

M_PLAYER runs a second instance of itself, the first one disassembles the source file and the second one reassembles the destination file.

The **default** convertion is **without sound**. If you want to **keep the original audio track**, you have to save it first *(as explained above)*. Notice that, when this is done, the button « `Add a sound` » automatically turns to selected.

Use one image out of 2__ images.

The default value is 1 to keep all the frames. But you can set 2, 3, or more to reduce the frame rate.

T = 2__ /200 sec for each image.

The time for each frame. Here 2/200 sec means 100 frames per second ! Too much. A value of 25 means 25/200 sec, this is 8 frames per second.

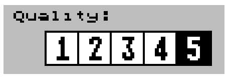
If you change the frame rate, you have to modify this timing accordingly.

If you use a **sound**, this value is **not used** as the frame rate will be computed to **fit the sound duration.**

One key frame every ___ images.

The key frames are a complete rebuilt of the screen. Other frames are only a modification of the previous image, they are called delta frames *(they store the difference)*.

If you want M_Player to be able to **skip frames** on **slow machines** to maintain the synchronization, you have to add key frames. One per second is not so bad. *(see Key-Frames use).*


Quality:
1 2 3 4 [✓ 5]

The quality setting tells M_Player the threshold from witch a pixel is considered as equal or changed compared to the previous image.

For 16 bits encodings :

Quality of 1 is the worst. A pixel has to notably change its color to be added to the new image.

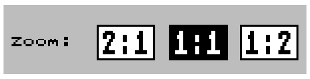
Quality of 5 is the best. A pixel has to be exactly the same as before to be skipped, else it is added to the new image.

So you can see that, with a low quality, few pixels will be stored in the delta frames : this will lower the size of the video and speed up the display.

On the contrary, a high quality will lead to larger files and slower replay.

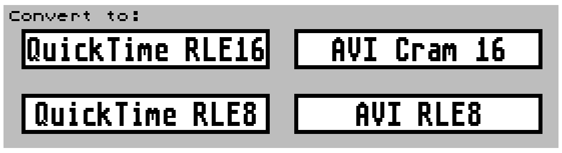
For 8 bits encoding :

Quality just tells to use color dithering (q=5) or grey scales (q=1).


Zoom: 2:1 [✓ 1:1] 1:2

You can keep the original size or zoom it by two (2:1) or reduce it by 2 (1:2).

When all parameters are ok, use one of these four exit buttons to start conversion :


Convert to:
QuickTime RLE16 | AVI Cram 16
QuickTime RLE8 | AVI RLE8

QuickTime RLE 16 :

fast encoding and replay, TC colors but low compression that leads to large files.

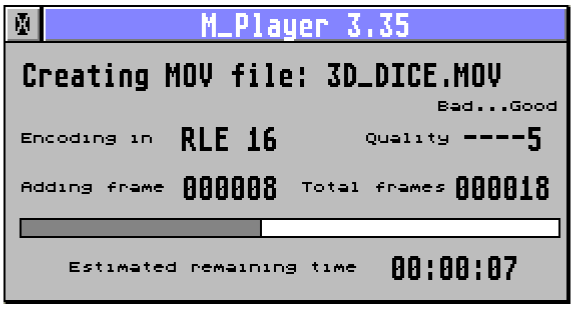
AVI Cram 16 :

slow encoding, average replay, TC colors and better compression.

QuickTime/AVI RLE8 :

fast encoding and replay, 256 colors with dithering or greys, average compression.

You’ll get a conversion box with bar progression and time estimation.



M_Player 3.35

Creating MOV file: 3D_DICE.MOV

Bad...Good

Encoding in RLE 16 Quality ----5

Adding frame 000008 Total frames 000018

Estimated remaining time 00:00:07

Note :

If the source animation has non standard width and height (odd values for example), this can lead to a bad display in the destination file.

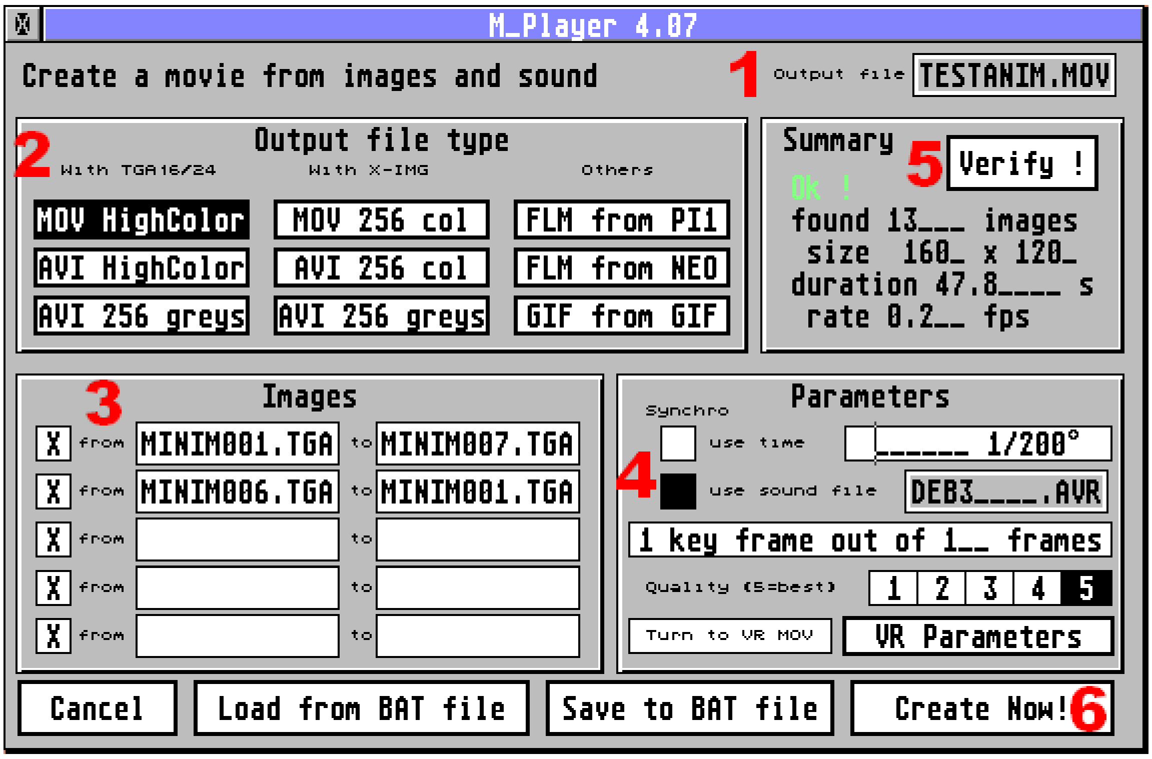
If this happens, then save the frames as TGA with New H and New L rounded to 4 or 8, and reassemble the animation.

V Build a video from images/Slideshows

M_Player allows you to put together a **set of images** and eventually a **sound** into a unique video file. Several formats are supported : **QuickTime, Video for Windows, Gif and FLM**.

Assembling a video **requires a batch file**. It's a simple text file with the informations needed to build an animation. You can create it by hand or use the old EASY_BAT.PRG tool, but the easiest way is to use the **integrated panel**.

V.1 The integrated panel



M_Player 4.07

Create a movie from images and sound — 1 — Output file: TESTANIM.MOV

2 — Output file type

With TGA16/24	With X-IMG	Others
MOV HighColor	MOV 256 col	FLM from PI1
AVI HighColor	AVI 256 col	FLM from NEO
AVI 256 greys	AVI 256 greys	GIF from GIF

Summary — 5 — Verify !
Ok !
found 13___ images
size 160_ x 120_
duration 47.8____ s
rate 0.2__ fps

3 — Images

X	from	MINIM001.TGA	to	MINIM007.TGA
X	from	MINIM006.TGA	to	MINIM001.TGA
X	from		to	
X	from		to	
X	from		to	

4 — Parameters
Synchro
☐ use time: ______ 1/200°
■ use sound file: DEB3____.AVR
1 key frame out of 1__ frames
Quality (5=best): 1 2 3 4 [✓ 5]
Turn to VR MOV — VR Parameters

Cancel — Load from BAT file — Save to BAT file — Create Now! 6

Here is the dialog in witch you'll **define your animation**. The steps are:

(1) Define the **output name** by clicking here. *The default path points to the "CONVERT" folder that you specified in Set Options and Paths.*

(2) Select the **output file type**. As I am using TGA files, I select MOV HighColor.

(3) Define the **images**, here two groups to get a back and forth movement between MINI001 and MINI007. *The default path points to the "IMAGES" folder that you specified in Set Options and Paths.*

(4) Set some **parameters**, here I don't specify a time for each image but I use a sound file. *The default path points to the "SOUNDS" folder that you specified in Set Options and Paths*. The frame rate will be computed according to the sound duration. Every image is a key-frame with the best quality.

(5) Clic on **Verify!** to make sure that everything is consistent (images with same sizes matching the anim file type, etc). You can see in green Ok! and the specifications of the anim that will be generated. If an error is detected, you'll see Errors in red.

(6) Clic on **Create now!**, the batch file is stored in memory and directly used by the program.

You can also **save the batch file to disk**, so you'll be able to reload it and modify some parameters to get a different video.

You can **modify the file by hand**, but **never edit** the lines **from DIAL to ENDD**. They contain every information to rebuild the panel as you left it.

Another example with Quick Time VR files is explained at QuickTime VR movies.

V.2 Limitations and links between image type and animation

- With **TGA 16b/24b** compressed or uncompressed, you can create :
  - ✔ a Quick Time RLE16 movie (MOV), High Color
  - ✔ a Video for Windows CRAM16 movie (AVI), High Color
  - ✔ a Video for Windows CRAM8 movie (AVI), 256 grey levels.

- With **GEM XIMG 256** colors images, you can create :
  - ✔ a Quick Time RLE8 movie (MOV), 256 colors
  - ✔ a Video for Windows RLE8 movie (AVI), 256 colors
  - ✔ a Video for Windows CRAM8 movie (AVI), 256 grey levels.

- With **GIF** images you can create a GIF movie.

- With **Degas PI1 or Neochrome NEO** images, you can create an Extended FLM movie.

V.3 Structure of a batch file, create it by hand.

You can get **more precise and powerful features** programming your own batch file with a simple text editor.

A batch file is an ASCII text files that contains 3 parts:
- the **identifier** with an optionnal DIAL/ENDD section
- the **header**
- the **datas**

The batch file can be used to display a slideshow of images. The only difference between slideshow and creation is the presence of a line `o=<file name>` that specifies the output file name.
Load this batch file into M_Player and use « **PLAY !** » to run the slideshow. You have to enable « **Save As...** » to initiate the creation if available (if `o=<file name>` is present).

Only the very first line must start ont the leftmost row, for the others, spaces and tab to indent are allowed, there must be only one command per line, lines starting with ';' are considered as comments.
The size of the file is limited to 12488 bytes (I think it's enough...).

The identifier

It must be on the first line, first character and is always:
`M_PLAYER` or `m_player` (or `m_PlAYer` if you prefer!)

Just following the identifier, the DIAL/ENDD section can be present if the file was saved from the integrated panel. Never edit those lines, or remove them all...

The header

Starts on the second line *(or after some comment lines)* and gives general infos on the slideshow:

`w=xxxx`	the width of the movie
`h=xxxx`	the height of the movie
`c=xxxx`	the compression (the type of file): `tga2` for TGA 2 uncompressed (24 or 16 bits) `tgac` for TGA 10 RLE encoded (24 or 16 bits) `ximg` for IMG with 'XIMG' extension and 256 colors `gif8` for either gif87a or gif89a `dega` for Degas 320x200x16 `neoc` for Neochrome 320x200x16
`b=xxxx`	size of the buffer to load the entire largest image file

Those first 4 infos can't be ommited! The following can (for a slide show, but must appear for a MOVIE-creation):

`t=xxxx`	time for one frame (number of 1/200sec) With a GIF it is the time to wait between two frames (without the display time), with 0, no delay block will be added.
`f=xxxx`	number of frames in the whole slideshow
`o=fpath`	path and name of the file to create (MOV, RLE16, RLE8 or GIF, or FLM)
`s=fpath`	path and name of the sound file (WAV or AVR, 8/16 bits, mono/stereo). For MOV/AVI the frequencies are 11025, 22050, 44100. For FLM 12517, 25033, 50066 +/-2% and only 8 bits.

If 's=' is present, then 't=' is ignored and the frame rate is computed according to the sound duration.

`q=x`	quality, default value is 5 (5 bits per color), but you can reduce it to 4,3,2 or even 1 to get a more efficient compression, but with a lost of quality. This parameter is used only in AVI and MOV files in 16 bits compressions. For 8 bits compression, a quality of 1 means grey levels !
`k=xxxx`	specifies the rate of the key frames (for example, with k=5, frames #5, 10, 15, 20 ... will be key frames). See below: "Key-Frames use". This parameter is used only in AVI and MOV files.

`r=xxxx` for GIF files only, fixes the number of loops for that movie, the range is 0 to 65535. This will add the NETSCAPE extension as follows:
$21, $ff, $0b, 'NETSCAPE', '2.0', $03, $01, Intel word=xxxx, $00.
A value of 0 indicated an infinite loop.

`v=max_x,max_y,start_x,start_y,img_cell,loop_x`
for MOV files only, this allows you to generate a QuickTime VR file (an interactive movie).
`max_x`=number of columns (start at 1)
`max_y`=number of lines (start at 1)
`start_x`=horiz. index of the first cell (0 to max_x-1)
`start_y`=verti. index of the first cell (0 to max_y-1)
`img_cell`=number of images per cell
`loop_x`=1 for horizontal looping, 0 to stop at boundaries
See 'QuickTime VR movies' below for more details.

`m=xxxx` if you call M_Player as a movie maker (MOV,AVI,FLM,GIF) from your own program and you don't want to fill your hard drive with TGAs, you can send to M_Player the images one by one using a kind of dialog through this stucture:

```
xxxx is an address that points to three longs:
     LONG your_routine
     LONG frame number requiered by M_Player
     LONG the adress where your pixels are
     WORD flag
     WORD extension
```

Each time my compressor needs an image file, it fills the frame number and then calls your routine. Your routine must fill the buffer address and the flag format.
Supported values are (for now):
`FLAG='EX'` then an extension is available
`extension = $0201` zoom×2 or `$0102` zoom÷2.
The FILE names given in the "data" section are dummy but must appear, for example, with 100 images you can write:

```
data
DUMMY.TGA
.rept 99
.disp
.endr
.stop
```

Then you call M_Player with:

```
m_player -d+a my_batch.bat
```

(no dialogs, Save as).

The order of those infos is not fixed, but you musn't put spaces between the characters :

`w=320` is ok
`w = 320` is wrong

For MOV and AVI creation, the lines « o= » and « q= » are used to specify the exact type of file you want.

Suppose we want to create a file named MYVIDEO with quality 4 **from TGA files** :

Output file type	Output file name	Quality
MOV RLE16	o=myvideo.mov	q=4
AVI CRAM16	o=+myvideo.avi	q=4
AVI CRAM8/grey	o=+myvideo.avi	q=-4

Suppose we want to create a file named MYVIDEO **from XIMG files** :

Output file type	Output filename
MOV RLE8	o=myvideo.mov
AVI RLE8	o=#myvideo.avi
AVI CRAM8/grey	o=+myvideo.avi

The data

Start immediately after the header with a key word:
`data`

Then, every line is read and:
if starting with a '.' then it's a command, executes it
else it's the name of a file, displays it and its name becomes the current name.

commands are:

`.rept xxxx`
`.endr`
repeat xxxx times what's between rept and endr. Loops can be nested.

`.disp`
display the file corresponding to the current name

`.getp`
use the palette of the next file (only applied with palette based images). If '.getp' is not specified, then the palette will be ignored (for example when all of the frames share the same palette that is loaded once), this speeds up the display when running in NOVA 256c (because the palette must be passsed to the VDI).
* not used with GIF movies *

`.incr`
increments a number found into the current name
Example, if name was: C:\IMAGES\FRM00001.TGA
it becomes C:\IMAGES\FMR00002.TGA.

`.decr`
same as above with a decrementation.

`.stop`
end of the slideshow. This command MUST appear, else the program crashes.

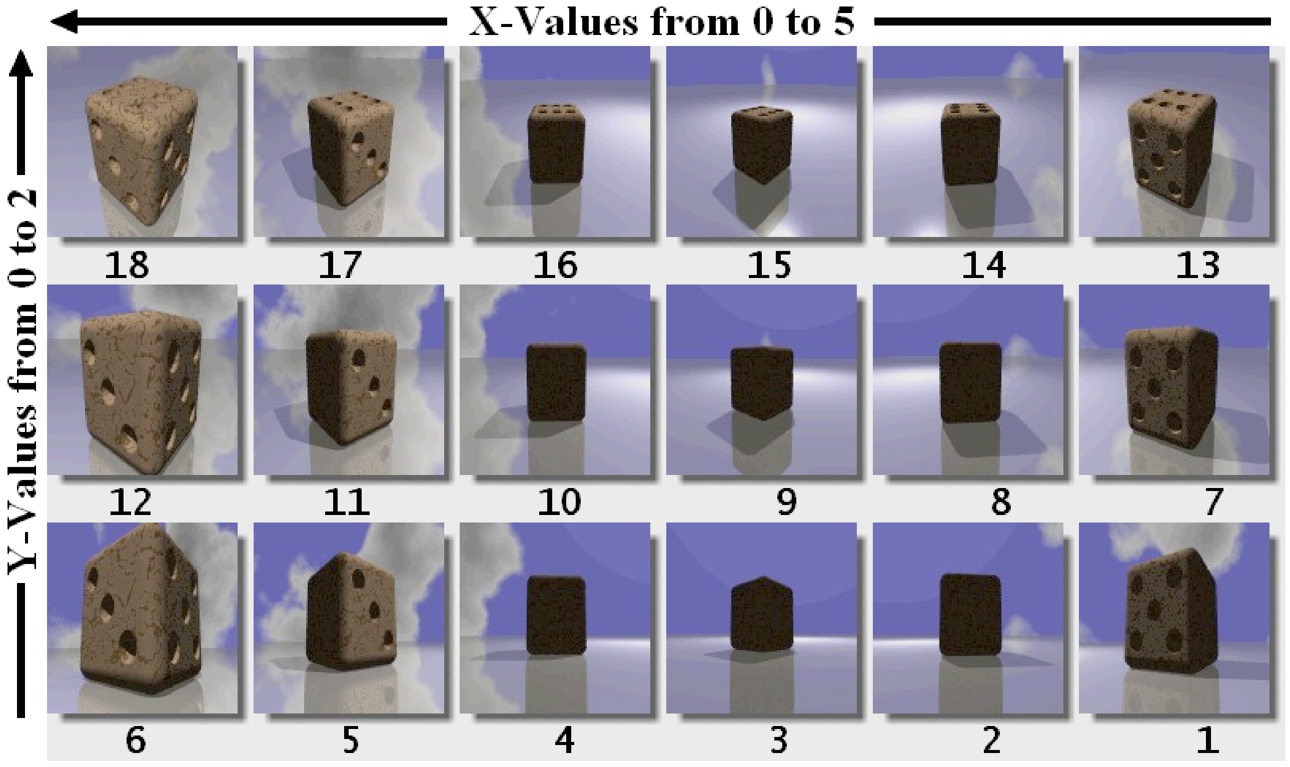
`.keyf`
forces the next frame to be a key-frame (if k=xxxx is used, then this auto key counter will be reset).
See below at "Key-Frames use".
* not used with GIF,FLM movies *

V.4 QuickTime VR movies

VR stands for **Virtual Reality**. Those movies are **interactive** and allow the user to navigate within a 2D-universe by changing the point of view of a camera horizontally or vertically or both. The navigation is available with the arrow keys or with the mouse.

Let's look at an example, I have 18 images of a dice. They are divided in three groups corresponding to the elevation of the eye :

- one group of six images viewed from the ground. (images #1 to #6)
- one group of six images viewed at mid-height (images #7 to #12)
- one groupe of six images viewed on top of the dice (images #13 to #18).



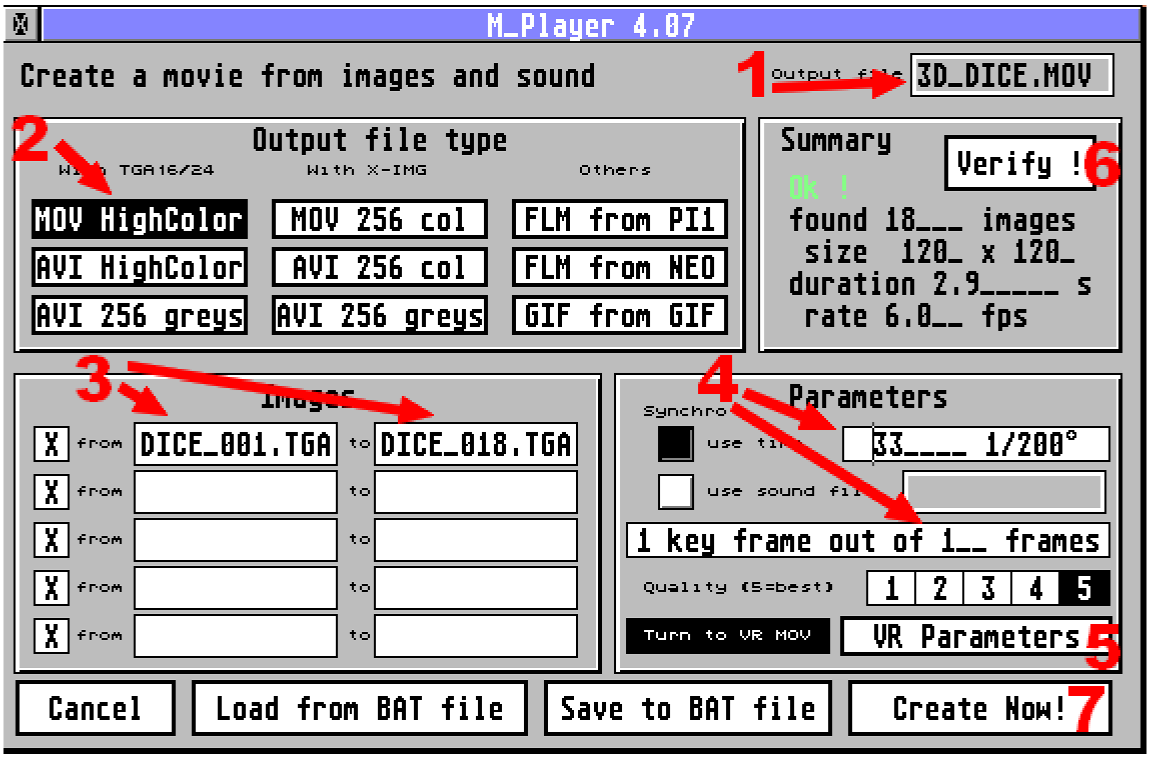
Fixed cells

When the movie starts, M_PLAYER displays the initial image, let's say that this one is at coordinates (X=3 ; Y=1) : this is image #10. Then, using the arrow keys :

- ➔ UP will display image #16
- ➔ DOWN will display image #4
- ➔ RIGHT will display image #9
- ➔ LEFT will display image #11

If the **horizontal loop** is set, then you can move from image #7 to image #12 for example.

These are the **settings** you have to make in the dialog:


M_Player 4.07

Create a movie from images and sound

1 Output file: 3D_DICE.MOV

2 Output file type

With TGA16/24: MOV HighColor, AVI HighColor, AVI 256 greys

With X-IMG: MOV 256 col, AVI 256 col, AVI 256 greys

Others: FLM from PI1, FLM from NEO, GIF from GIF

Summary — Verify ! 6

Ok !
found 18___ images
size 120_ x 120_
duration 2.9_____ s
rate 6.0__ fps

3 Images

X	from	DICE_001.TGA	to	DICE_018.TGA
X	from		to	
X	from		to	
X	from		to	
X	from		to	

4 Parameters

Synchro
■ use time: 33____ 1/200°
☐ use sound file:

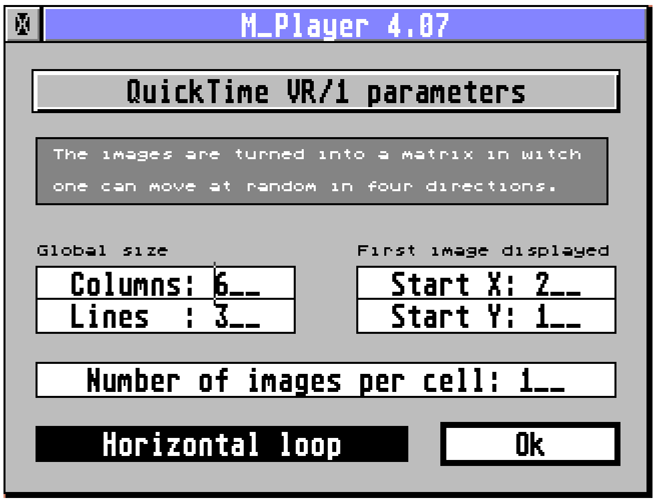
1 key frame out of 1__ frames

Quality (5=best): 1 2 3 4 [✓ 5]

Turn to VR MOV — VR Parameters 5

Cancel | Load from BAT file | Save to BAT file | Create Now! 7

Don't forget to clic on **<u>VR Parameters</u>** and fill in the following informations:


M_Player 4.07

QuickTime VR/1 parameters

The images are turned into a matrix in witch one can move at random in four directions.

Global size
Columns: 6__
Lines : 3__

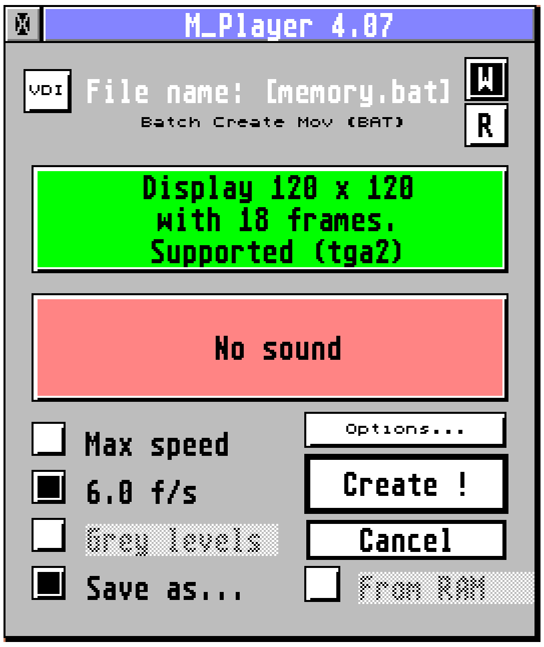
First image displayed
Start X: 2__
Start Y: 1__

Number of images per cell: 1__

Horizontal loop | Ok

With a value of **one image per cell**, this VR wil display fixed images that you'll animate with the mouse or the keyboard.

Then, back on the main dialog, clic on **<u>Verify !</u>** to ensure that everything is correctly set.


M_Player 4.07

VDI File name: [memory.bat] W R
Batch Create Mov (BAT)

Display 120 x 120
with 18 frames.
Supported (tga2)

No sound

☐ Max speed
■ 6.0 f/s
☐ Grey levels
■ Save as...

Options... | Create ! | Cancel | ☐ From RAM

You can save the BATCH file for future modifications.

Finally, clic on **<u>Create now !</u>** to prepare the video encoding.

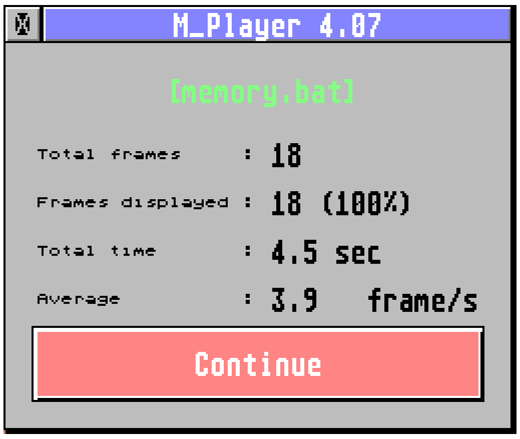
The dialog on the left summarizes your work, an internal BATCH file has been created (memory.bat) and will be used to generate 3D_DICE.MOV.

Clic on **<u>Create !</u>** and the work starts.

A progression bar appears and tells you how far you are.

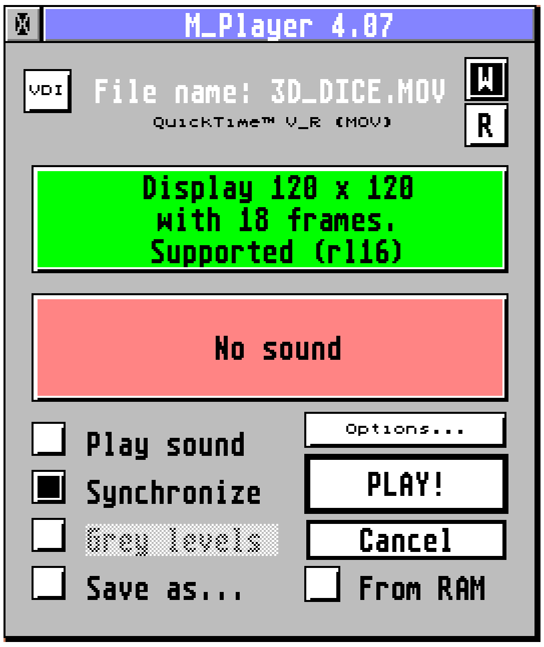
A stats box appears at the end.

Clic on **Continue** and you go back to the General Menu where you'll clic on **"Load and Play"** to load your 3D_DICE.MOV new animation.


M_Player 4.07

[memory.bat]

```
Total frames     : 18
Frames displayed : 18 (100%)
Total time       : 4.5 sec
Average          : 3.9   frame/s
```


M_Player 4.07

File name: 3D_DICE.MOV

QuickTime™ V_R (MOV)

Display 120 x 120
with 18 frames.
Supported (r116)

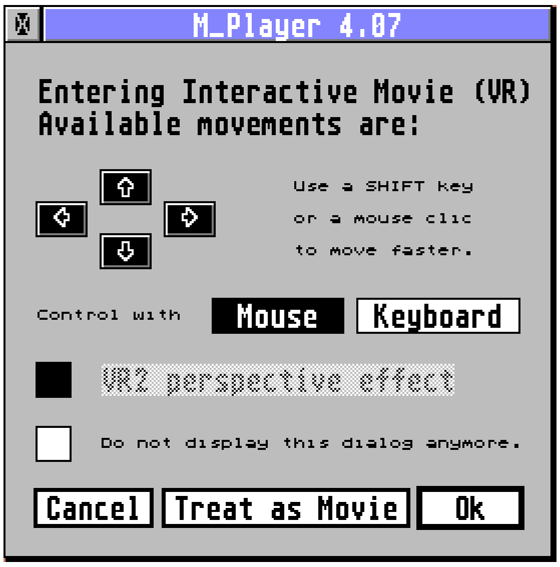
No sound

☐ Play sound
☒ Synchronize
☐ Grey levels
☐ Save as...
☐ From RAM

As you can notice, it is detected as a Quick Time VR animation.

Clic on **PLAY!**.

A new dialog appears, **specific for VR** files:


M_Player 4.07

Entering Interactive Movie (VR)
Available movements are:

Use a SHIFT key
or a mouse clic
to move faster.

Control with [✓ Mouse] Keyboard

☒ VR2 perspective effect

☐ Do not display this dialog anymore.

You can select the **control method**, either using the **mouse** or the **keyboard**.

You can treat the file as a standard movie (the 18 images will be displayed in sequence), but if you clic on **OK**, you enter the VR mode.

Now you can observe your dice up and down, and all around!

If the speed of 6 frames/sec is not enough, just **press SHIFT** while you are using the mouse or keyboard and the frames will be displayed **with no delay**.

The corresponding BAT file to create the Virtual Reality movie is :

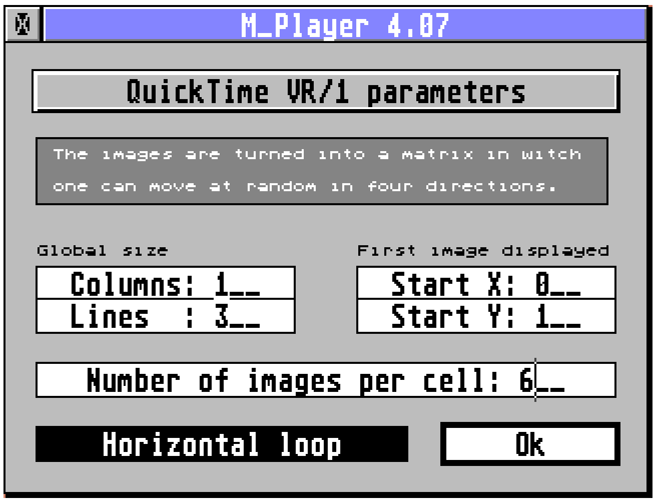
```
M_PLAYER                 ** identifier
c=tgac                   ** TGA compressed images
w=120                    ** width
h=120                    ** height
b=18944                  ** buffer for max image
o=K:\VR\3D_DICE.MOV      ** output file
q=5                      ** best quality
k=1                      ** all are key frames
t=33                     ** 6 image / sec for one turn around
f=18                     ** 18 frames
v=6,3,2,1,1,1            ** 6 columns, 3 lines, start in (2;1)
                         ** 1 image per cell and loop allowed
data                     ** start data section
K:\VR\DICE_001.TGA       ** first image
.rept 17
.incr                    ** add the 17 following
.disp
.endr
.stop                    ** end.
```

Important note :

Every frame must be a key frame as the order of the display is not predictible. See at Key-Frames use.

See EASY_BAT.PRG usage for a complete description of the BAT creation.

Animated cells


M_Player 4.07

QuickTime VR/1 parameters

The images are turned into a matrix in witch one can move at random in four directions.

Global size

Columns: 1__

Lines : 3__

First image displayed

Start X: 0__

Start Y: 1__

Number of images per cell: 6__

Horizontal loop

Ok

One **cell can contain an animation** instead of a fixed image.

For example, see the images above, you can imagine only three vertical cells and each one contains six views of the dice that are automatically animated. So you can just move up and down and the horizontal movement is automatic.

Just modify the VR Parameters as shown on the right. Or, if you work with a text editior, just change one line :

```
v=1,3,0,1,6,0            ** 1 column, 3 lines, start in (0;1)
                         ** 6 images per cell no loop allowed
```

Panoramas

These are the second form of VR files where things are inverted. Your are not looking at an object around it, but you are inside a place and you can rotate around this center point to explore all the place.

M_PLAYER can display such images but has not the ability to reinder the correct proportions of what is seen.

V.5 GIF movies

A **GIF movie** can only be created **from GIF images**, those images must share the same global size and the same global palette (if a local palette is present, it will be stored).

Every other bloc than $2C (the images) is skipped *(for example copyrights, infos, etc...)*. Only one bloc is always added, the one defined by myself that indicates the number of frames:

```
$21, $ff, $0b, 'M_PLAYER', 'FRM', $04, Intel Long Frames, $00
```

This bloc must be located just after the local palette *(and, if present, after the Netscape extension concerning the looping).*

The **delay time** between two frames is stored into an unsigned word, so it is limited to 65535, this means no more than 655.35 seconds *(and so 131070 for 't=' as it is a number of 1/200 of second)*. If the value is greater than this limit, 0 will be used instead (meaning no delay!).

M_Player can be **used to modify an existing GIF-Animation** *(for example to add the M_PLAYER extension, the Netscape extension or to modify the delays between two frames,...)*.

Example :

I have an animation NUKE01.GIF that has no M_Player extension, with a infinite loop and that is played too fast, I run it once to get the infos:

size 64x64, 9 frames displayed (in the stats box at the end. When an anim has a loop, SHIFT stops it!).

Then I create the following batch file:

```
m_pLaYeR
w=64                the size I noted
h=64
c=gif8
b=16000             the file is only 15xxx bytes long
t=11                11/200 = 0,055s of delay
f=9                 9 frames into my initial anim
r=3                 will repeat only 3 times (instead of infinite)
o=C:\TGA\nuke.GIF   the output file
data
C:\TGA\nuke01.GIF   only one file!
.stop
```

Upon exit, I'll have a **new anim** NUKE.GIF with the **M_PLAYER extension**, with different delays and with 3 loops only. The output file is shorter, because M_Player has skipped every info bloc!

V.6 FLM Movies

As they are created from ST-Low images (c=dega or c=neoc), you can omit 'w=', 'h=' and 'b=', they will be filled with the following default values:

```
w=320
h=200
b=33000 (PI1 32034 or 32066, NEO 32128)
```

A FLM file **hasn't the 'key frame' capability**, and so, be sure that the frame rate you specify will be possible on your machine. Else, as no frame is skipped, the images could last longer than the sound.

They are disigned for the Atari Video system. The encoding of the images is exactly the same as the screen. So, the decoding is done directly in video RAM with a hight speed even on a simple Atari ST. I extended this format with sound for the Atari ST/STE, it is largely used in MP_STE.PRG, the video player for Atari 68000.

V.7 Example of a BATCH FILE

Example of slideshow :

(The comments starting with ** don't appear in the file!)

```
** start of file
M_Player                    ** Id
; comes from a FLI          ** comment
w=320                       ** width
h=200                       ** height
; 67/200 means 3 f/s        ** comment
t=67                        ** time for one frame
f=29                        ** number of frames
c=tga2                      ** files are TGA 2 uncompressed
b=192018                    ** size of one file (320x200x3 + 18)
data                        ** end of header/start of data
E:\imprimer\mou00001.tga ** first frame, display it
.rept 2                     ** twice the whole
  .rept 7                   ** 7 times
    .incr                   ** increment name of file (00002 to 00008)
    .disp                   ** and display it
  .endr                     ** loop
  .rept 7                   ** another 7 times
    .decr                   ** dec name of file (00007 to 00001)
    .disp                   ** and display it
  .endr                     ** loop
.endr                       ** twice!
.stop                       ** end of slideshow
```

Well, this example has displayed:
mou00001.tga
mou00002.tga
...
mou00007.tga
mou00008.tga
mou00007.tga
...
mou00001.tga
And this twice! The images came from a FLI that I saved using the Step by Step mode.
INCR and DECR commands can only work correctly if the names of the files END with digits! (for example 00001BB.TGA won't work because it ends with BB).

Example of creation :

You can specify each file if you don't want to use commands:

```
M_PLAYER
w=184
h=240
b=50000
c=ximg                      ** t= ommited because there's a sound
f=5
o=#C:\ANIM\US.AVI           ** will create an AVI ('#')
s=C:\SOUNDS\KISS.AVR        ** will add this sound
data
.getp                       ** supposing that all have the same
C:\IMAGES\I.IMG             ** palette, we load it once!
C:\IMAGES\YOU.IMG
C:\IMAGES\I.IMG
C:\IMAGES\YOU.IMG
C:\IMAGES\YOU_N_I.IMG
.stop
```

'.stop' is the only command that must appear.

Common errors (those I made...)

- ✗ The header of the output file is built BEFORE adding the frames, so the **number of frames** that you specify (f=xxxx) **must be exact**! Else you'll get a totally corrupted file. To ensure that it's the right value, launch a slide-show before encoding, if total_frames=frames_displayed, it's Ok.
- ✗ This is the same for 'w=' and 'h='. Be sure that **all of the images have the same size**, the one you specified.
- ✗ The 'supported' message into the graphic box is based upon your 'c=xxxx'. The program hasn't read a file to verify that, ensure that it's the good fromat.
- ✗ The **sound frequencies** accepted are those from the PC world: **11025, 22050 and 44100** KHz, or the closest possible to them. Else, you get an 'Unsupported' message in the sound box.
- ✗ If using IMG images you get the RLE8 compression that uses a palette. But, the output file can contain only ONE palette, so you must be sure that **all the images share the same colors**, else, it's the last palette read (with .getp) that will be saved.

V.8 Key-Frames use

Here is a brief introduction to the relevance of **inserting key-frames** into a movie. There are two cases:

- your **machine is fast** and no frame will be skipped when playing, then you don't need to use the key frames.

- your machine **isn't so fast** and there may be some frames skipped, the use of key-frames is highly recommended!

What happens when a frame is skipped?

In order to **increase the compression**, only the **differencies between two frames** are stored, this means that one frame is only a **partial display**. If one frame is skipped *(because of the synchronisation)* the next one won't have the expected background! You'll get a wrong display.

A **key frame** is used to solve this problem, it is a complete rebuilt of the screen. So, for synchronization needs, if M_PLAYER has to skip frames, it jumps to the next key-frame to go on displaying.

<u>An example:</u>

Suppose you have used k=5 and t=40 *(5 frames per second)* into the header and that your machine can only display 3 images /second. Here is what will happen:

second #1: frames 0,1,2
second #2: frames 5,6,7
second #3: frames 10,11,12
etc...

Every key frame will be at its exact position *(in time)*, and between 2 key-frames, the player does as much as it can. If you play this same movie with a faster machine, you'll get maybe:
second #1: frames 0,1,2,3
second #2: frames 5,6,7,8
second #3: frames 10,11,12,13
etc...
And with a very fast machine, you'll get every frame.

The best is to put 1 or 2 key-frames per second, this way, the display will be synchro with the sound once or twice a second.

You must know that a key frame **is larger** than a normal one, because it rebuilds every pixel, and that inserting too few key-frames will lead to a poor synchro quality. It's a **question of balance**. Make some tests to find the best solution.

<u>Idea:</u> *it's a good idea to choose the frames where the background changes suddenly, because anyway most of the pixels will be rebuilt!*

VI Albums

With this feature you can **group up to nine videos** into one album with a **short description** for each. For example, this can be a **thematic album**: songs, cartoons, family, etc.

Or a group corresponding to a **physical support** such as a CD-ROM. This way, you don't have to browse the disk looking for hidden places with animations, they will be easely reached.

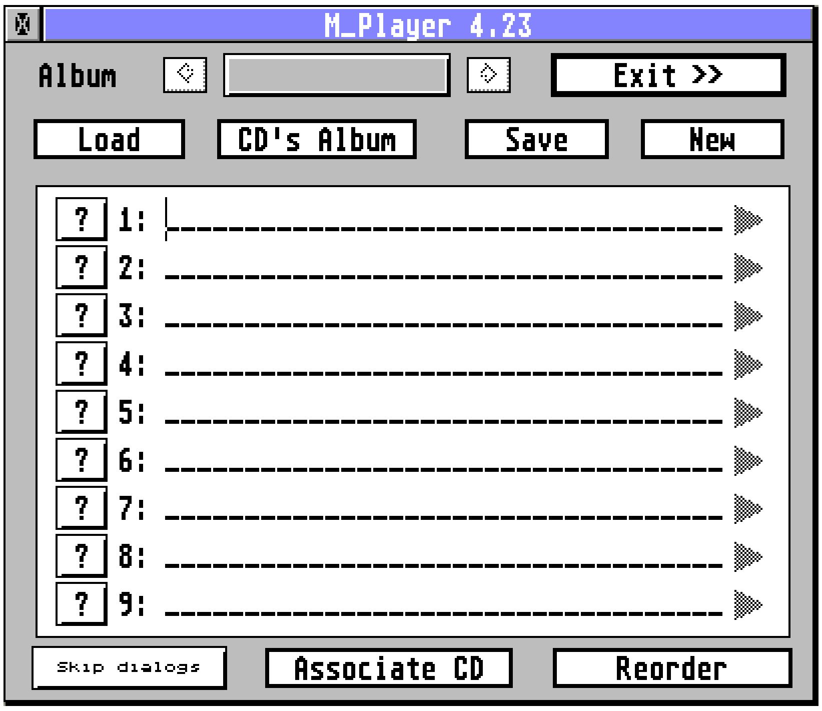
VI.1 Create an album by hand

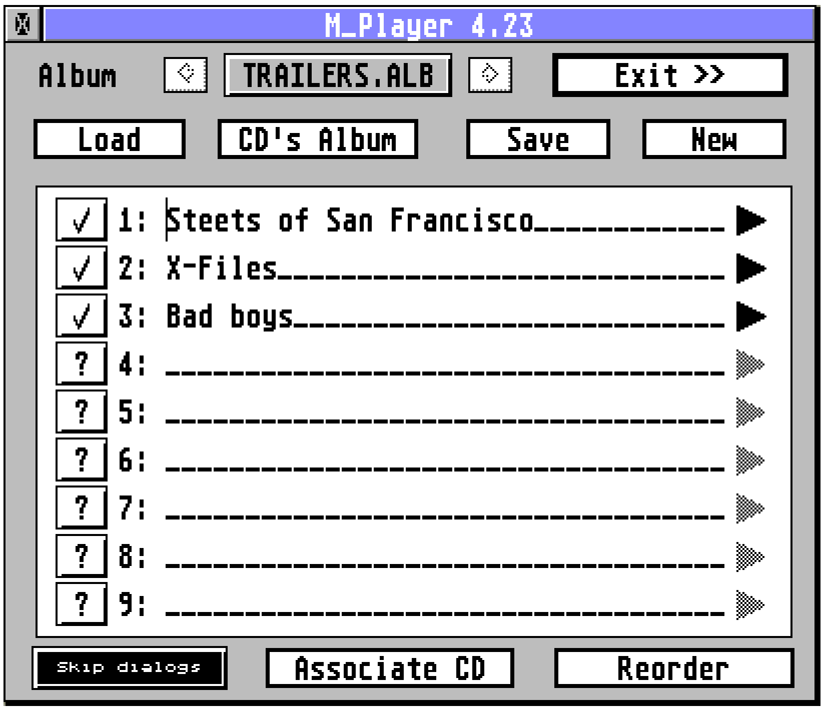
If it's the first time you enter Albums, an empty dialog appears, else clic on "New" to get a brand new page.

To define a video entry, you must clic on one of the nine "?" buttons.

This opens a filselector and you can define your video.

If you validate this, then the "▶" button is enabled and the cursor is placed on the description line, so you can put a short comment.





✓ 1: Steets of San Francisco
✓ 2: X-Files
✓ 3: Bad boys
? 4:
? 5:
? 6:
? 7:
? 8:
? 9:

Notice that that "?" is changed into "☑" when a video exists.

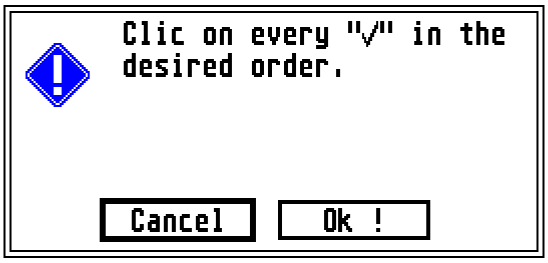
Load, Save

allows you to load and save your album. In this case its name appears in the upper box.

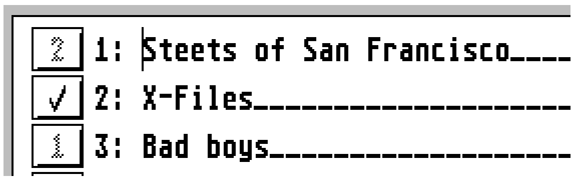
Reorder

If you have more than one video in your list, you can reorder then. If not, you'll hear a bell sound meaning reordering is not available.

An alert box tells you how to reorder.


Clic on every "√" in the desired order.

Suppose I want the previous list to appear in alphabetical order. Then I have to clic first on "☑" for "Bad boys", notice that it turns to "1" and becomes disabled. So you can't clic on it anymore.



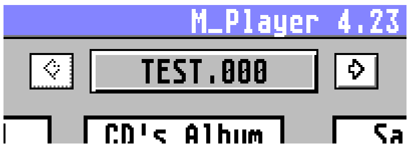
Then on "☑" for "Streets of San Franciso", it turns to "2" and becomes disabled.

The last one will be X-Files. The list automatically appears in the order you wanted.

Don't forget to save the Album if you wish to keep it reordered.

VI.2 Create an album automatically

You can use the Search a disk for animations facility to create albums. If more than nine files are found, then more than one album are generated and they are automatically chained the one to the other.



That's the goal of the two arrow buttons you can see around the Album's name.

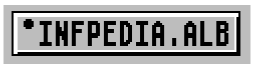
In the example on the right, the file TEST.000 is chained to another file, clic on the right arrow to display the content of TEST.001.

VI.3 Linking an album to a CD-Rom

Associate CD / Dissociate CD

Wether you created it by hand or automatically, if your set of videos come from a CD-Rom, you can mark this album with the CD signature using "Associate CD". M_PLAYER will automatically find the CD drive unit, and take some informations to sign you album.

Note that when an album is associated with a CD disk, a black spot appears in front of its name:



If you want to remove this link, just use "Dissociate CD".

CD's Album

That's the way to use the link ! Just insert the CD in your drive and clic on CD's Album. M_PLAYER opens a fileselector, just go to the album folder and validate. No file name is required as the program will find automatically the album that was associated to this CD.

VI.4 Using an album

▶ (play)

Plays the video. The default setting is to show the dialog boxes. The main one before playing with the CODECS informations and the statistics at the end. After a replay, the program goes back to the album. To exit to the General Menu, use "Exit >>".

Skip Dialog

If you clic on this, then the replay will be done without any dialog box.

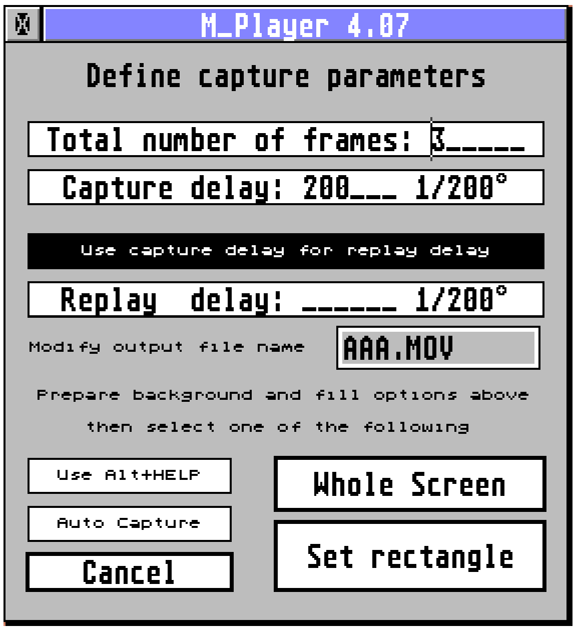
This preference is saved inside the album file.

VII Capture the screen to an animation

You can use M_PLAYER to grab the screen of another application (either as an accessory under TOS or as an application under Multitasking) to build an animation on the fly.

The type of animation depends on the current resolution:

- if your screen is in **high-color or true color**, you'll get a **16 bits compression** (MOV RLE16 or AVI CRAM16)
- if your screen is in **256 colors** or less, you'll get a **palette based compression** (MOV RLE8 or AVI RLE8)



M_Player 4.07

Define capture parameters

Total number of frames: 3_____

Capture delay: 200___ 1/200°

Use capture delay for replay delay

Replay delay: ______ 1/200°

Modify output file name AAA.MOV

Prepare background and fill options above then select one of the following

The dialog appears to let you set your parameters.

Be sure to **fill the upper part before** using "Whole Screen" or "Set rectangle".

- The **number of frames** will be the number of screen captures that will be added to the animation.
- The **capture delay** is the time in 1/200°sec between each capture when you use **Auto Mode**, else it is ignored.
- The **replay delay** is the time in 1/200°sec that defines the frame rate of the animation.
- The **output file name** is the name of the created movie.

VII.1 The manual mode

There are two ways to grab the screen:

- a hot key **Alt+Help** (Use Alt+HELP enabled)
- a **dialog** (Use Alt+HELP disabled)

The **Auto Capture** is discussed below.

Alt+Help method should be reserved for applications that don't give access to the menu and use the whole screen, in this case you can't reach your button to capture.

Another problem: *the Alt+Help hot key is not managed anymore on newer systems. It runs under TOS, again with Mint 1.12 but* ***has been abandonned somewhere between 1.12 and 1.19.***

Once you have selected the method clic on:

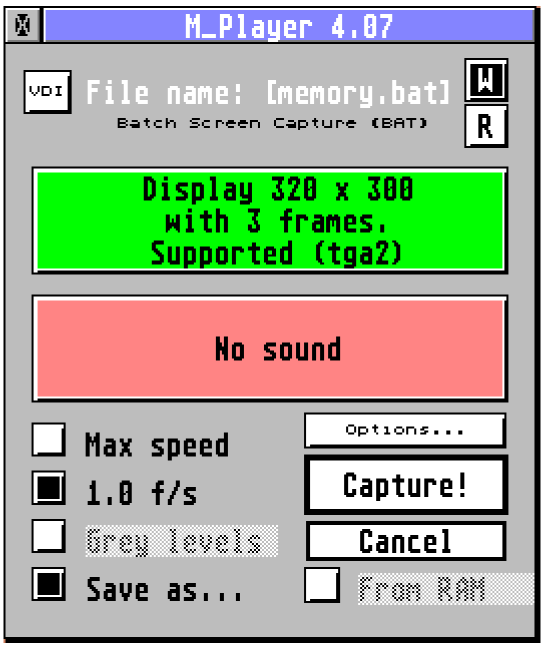
- ✔ **Whole Screen**, the capture will be performed on the whole screen including the menu.
- ✔ **Set Rectangle**, the cursor turns to a cross and you can clic and drag the rectangle to define your zone of capture.

When the rectangle is defined, you get the standard information panel with the chosen dimensions and the number of frames.

Note that **<u>Save as...</u>** is forced to validate the creation of an animation.

You can check the dimensions of your rectangle. *Important: the width is aligned on a 16 pixels boudary and that the height is aligned on a 4 pixels boundary.*

Just click on **<u>Capture!</u>** to start capturing.


M_Player 4.07

VDI File name: [memory.bat]
Batch Screen Capture (BAT)
W
R

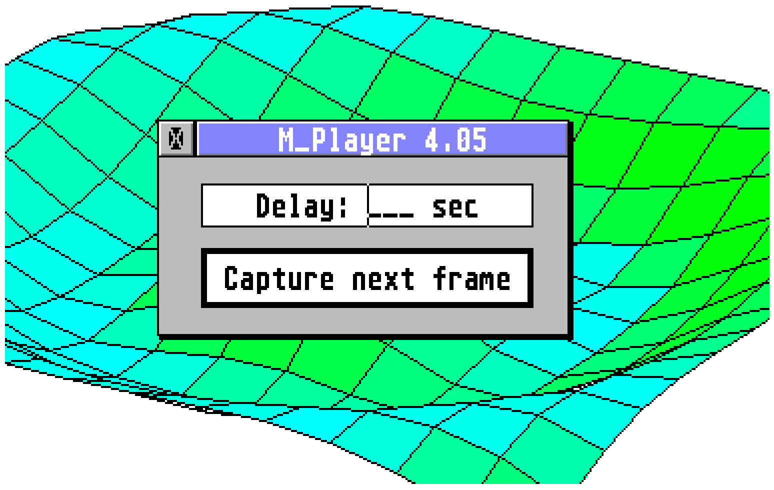
Display 320 x 300
with 3 frames.
Supported (tga2)

No sound

☐ Max speed
☒ 1.0 f/s
☐ Grey levels
☒ Save as...
Options...
Capture!
Cancel
☐ From RAM

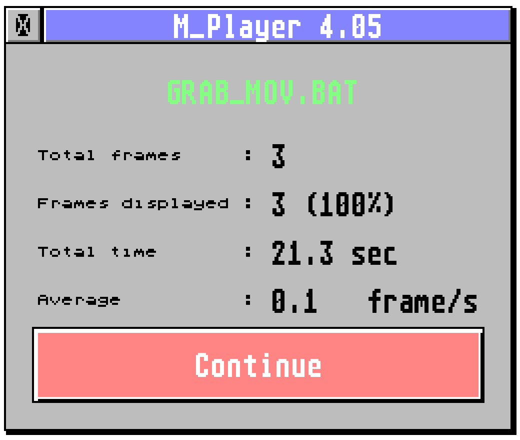
➔ If you are **using the Alt+HELP** method, a bell sound indicates when M_PLAYER is ready to grab the screen. Use your hot key when you are ready, and wait for the next bell. And so on.

➔ If you are **not using the Alt+HELP** method, then a little dialog appears to capture the screen. It includes a delay parameter, so if you think that the application will require 10 seconds before displaying the screen you want to capture, then set Delay = 10 seconds and clic on **<u>Capture next frame</u>**. And so on. You can specify a new delay for each frame.


M_Player 4.05
Delay: ___ sec
Capture next frame

When the screen is captured you can see a shrinking box appear. The time between this little effect and the next bell sound is the time taken by M_PLAYER to compress this screen and add it to the animation.

When all the planned frames are gabbed, then you get the final statistics dialog:


M_Player 4.05

GRAB_MOV.BAT

Total frames : 3
Frames displayed : 3 (100%)
Total time : 21.3 sec
Average : 0.1 frame/s

Continue

So you can to load your animation from where it was saved and see your work!

With a simple TT with no graphic card, the animation is correctly saved but remember that it is replayed using the TT Low resolution (320x480x256). Then if your animation exceeds the 320 pixels, it will be displayed as an approximation.

But as soon as you get a better display, the images will appear fully as they were captured.

VII.2 The auto capture mode

In the dialog **"Define capture rectangle"**, you can enable **"Auto Capture"**. This doesn't change the way you start your capture: either with Alt+Help or with the dialog using an optional delay.

But then, **as soon as the first frame is captured**, M_PLAYER enters in **automatic mode** and captures periodically the rectangle until the last frame.

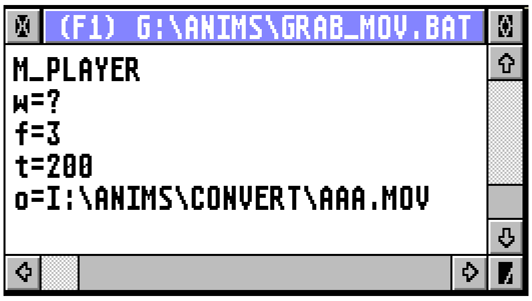
Be sure to set a **Capture Delay** that **matches** your **machine power**. If it is too short, then M_PLAYER will still be encoding a frame when the new one is grabbed. This can lead to a total mess in the system...

VII.3 Capturing using a batch file

Again, a BATCH file can be used to capture the screen! That's what is done internally when you use the standard dialog.

Anyway, you have to prepare a batch file. In this case, very few parameters are required:

- The identifier
- The number of frames (f=)
- The time for one image in 1/200 (t=)
- The output file name (o=)



```
M_PLAYER
w=?
f=3
t=200
o=I:\ANIMS\CONVERT\AAA.MOV
```

The key parameter to run a screen capture based animation is **"w=?"** meaning that you'll define the rectangle at run time with the mouse.

As soon as you have loaded this BAT file, you'll be driven to the previous dialog but the parameters will be filled for you according to what is found in the file.

If your BAT file contains the line:

```
t=val
```

Then the **same timing** is used for both **capturing and replaying** the animation. But you can separate those two behaviours with:

```
t=val1,val2
```

The **first** value will be the **replay** timing and the **second** the **capture** timing.
Example:

```
t=80,400
```

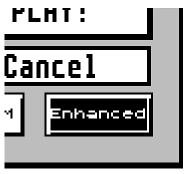
The screen will be captured every 400/200 sec = every two seconds, but the animation will be replayed with 80/200 = 0,4 sec, this is 2,5 frames per second.

VIII Enhanced files

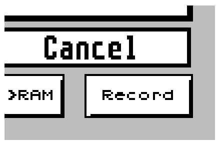
You can enhance the display by using a ***.ENH file** containing **subtitles or comments** and that will be displayed **under your video**.

The file must be located in the **same folder**, with the **same name** and **ENH extension**.

For example, to enhance MY_VIDEO.AVI, use MY_VIDEO.ENH.



If such a file is detected, then it is **loaded and compiled**. If everything is okay, then the button "Enhanced" appears selected into the main dialog box. You can uncheck it if you don't want the comments. See Enh button.



If the button appears as Record, this means that no ENH file was found or that it contains errors *(in this case, an alert box should tell you what's wrong with it)* and that **you can record one ENH file**. See at Recording an Enhanced file

If the button appears **disabled**, then no window mode is available *(for example when the computer needs to change the resolution to display the video)* and you can't use the enhanced features.

VIII.1 The header

The ENH file is a plain text file with one command on each line. The header contains one to three lines:

ENHANCED	-> the very first word of the very first line.
l=*(value)*	-> number of text lines under the video, if ommited, then l=1.
s=*(value)*	-> font size from 0 (small) to 3 (large), if ommited, then s=2.
c=*(value)*	-> color index from 0 to max colors (1, 15, 255), default is 1. Set a negative value to use this index only in 256 colors mode with palette change.

VIII.2 The different versions

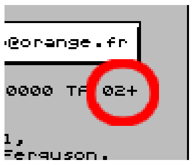
You can build a **multilanguage file** that will be automatically adapted to the current system language. If no version matches the system, then the very first version is used. So it is recommended to put something that most people will be able to read.

Each version start with:

v=*(value)*	-> the value is the language code (0=english, 1=german, 2=french...)

If two sections share the same **v=n** command, then the last one is used. This is useful to test your file, suppose your system is french and that your french version is the first one. Then you can test every other version by writing v=2 before it.

If you're not sure of your **country code system**, then have a look at the **credits dialog** box, the code appears at the end of the line in small characters:



VIII.3 The events during replay

They are a mix of time markers and text functions.

A time marker is a simple number in 1/100 sec. The time markers must appear in chronological order as the file is read sequentially.

840	-> what follows will be performed after 8,40 seconds.
d=*(text)*	-> the <u>text</u> til the end of the line is <u>displayed</u>. If you selected more than one line, then you can use more than one d=(text) command. At the next time marker, the display starts again at the first line.
cl	-> <u>clear</u> the text zone
be	-> output a <u>bell sound</u>
a=*(code)*	-> set <u>horizontal text alignment</u>. Available codes are:

a=l for left	a=c for center	a=r for right

e=*(codes)*	-> set <u>text effects</u>. You can combine several codes, available are:

n = normal	b = bold	i = italic
u = underlined	o = outlined	l = light
s = shadowed (not always available...)		

pa	-> enter pause mode. See Pausing.

VIII.4 An example

Here is an example with comments. The comments can appear in the file itself as after every command, the remaining characters of the line are ignored.

An exception is the **d=** command that displays everything til the end of line, so don't put comments on them!

`ENHANCED`	*the magic word*
`l=2`	*two lignes*
`s=2`	*font size = 2*
`c=1`	*text color = 1 (black)*
`v=0`	***English** version*
`100`	*1st event at 1,00 sec*
`a=c`	*center*
`d=Hello !`	*display those two lines*
`d=This is a second line.`	

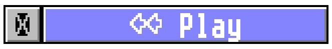
`200`	*then at 2,00 sec...*
`cl`	*... clear the messages!*
`350`	*then at 3,50 sec...*
`d=How are you?`	*... display this line*
`e=i`	*change the effects to italic*
`d=Fine thanks!`	*and display this second line*
`500`	*finally at 5,00 sec...*
`cl`	*... clear the messages. No more events.*
`v=2`	***French** version starts here.*
`100`	*all the same with french text.*
`a=c`	
`d=Salut !`	
`d=Voici la 2e ligne.`	
`200`	
`cl`	
`350`	
`d=Comment allez-vous?`	
`e=i`	
`d=Bien merci!`	
`500`	
`cl`	

VIII.5 Pausing

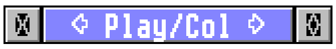
You can enter the **pause mode** either with the **"pa"** command of an Enhaced file or using the **Alt+Control** keys with any other animation *(be sure to press Alt first, else you quit the replay!).*

If the display is done **without window** *("W" button unselected or after a resolution switch or full screen required),* the only thing you can do is resume the replay by **pressing a key**.

If a **window is present**, then you can **interact with the desktop**. You can move the window, select menu entries, work on other applications. To **resume the replay**, just clic on the "Closer" button of the window.



In **256 colors modes**, the colors must be swapped between desktop settings and the animation requirements. On newer systems (as AES 4.00) M_PLAYER is notified if its window is topped or untopped, so it can swap the palettes. On older systems, you'll notice that the "Fuller" button appears and allows you to swap manually the palettes if they don't...



If you use the TT Low resolution (256 colors, 320x480), then moving the window is automatically forced on a 16 pixels boundary because of the planar encoding.

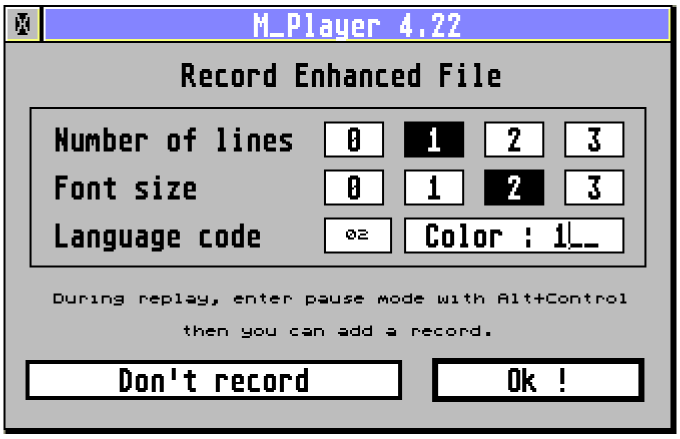
VIII.6 Recording an Enhanced file

The Record button allows you to **create automatically your ENH file** without having to write it by hand. Just after clicking on the Play ! button, this dialog appears and lets you define your general options.

Don't forget that if you're not satisfied with this, you can then edit the text file to meet your needs.

For example, you may want more than three lines.

Or you'll want to add another language by a copy paste operation.



Record Enhanced File

Number of lines 0 [✓ 1] 2 3

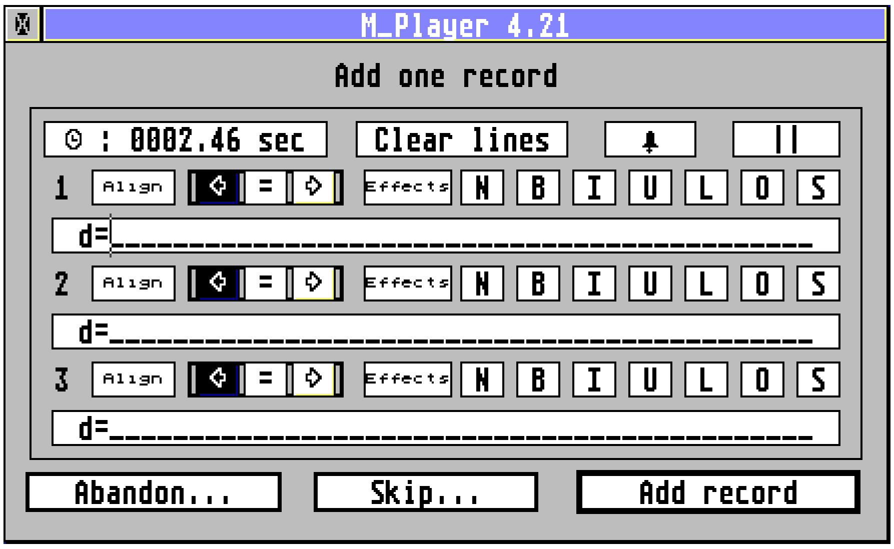
Font size 0 1 [✓ 2] 3

Language code 02 Color : 1

During replay, enter pause mode with Alt+Control then you can add a record.

Clic on Ok ! and the replay starts.

Use **Alt+Control** to enter pause mode. This new dialog appears.



Add one record

⏲ : 0002.46 sec

1 d=

2 d=

3 d=

On the very first line, the time appears and you can select to:

- clear lines **(cl)**
- add a bell sound **(be)**
- add a pause **(pa)**

Then, **three lines of text can be edited**.

- If you want to change the **alignment**, select Align and left, center or right.
- If you want to change the **text effects**, select Effects and some of the seven options.
- Fill the **text line** If a new line covers the previous one, you don't have to clear the lines, this can save some time at replay!

Remember that once you have set an alignement or an effect, the settings remain the same until you change it again.

At then of the video, your ENH file is ready and you can reload the animation to view it.

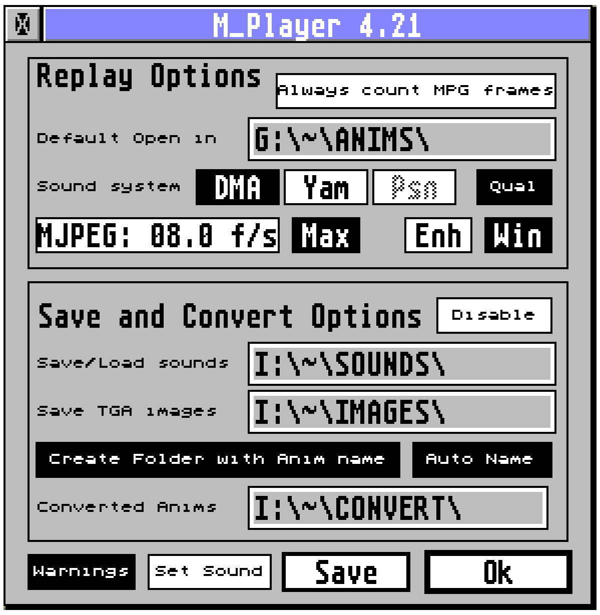
Note: as long as a **valid Enhanced file exist**, you **can't reach the** Record button.

IX Set Options and Paths

When in the main dialog box of an animation, or from the **General Menu**, you can enter the Options panel.

This one will ease your work as you'll be able to define several paths to avoid the use of many fileselectors while working.

IX.1 Replay Options


M_Player 4.21

Replay Options — Always count MPG frames

Default Open in: G:\~\ANIMS\

Sound system: DMA Yam Psn Qual

MJPEG: 08.0 f/s Max Enh Win

Save and Convert Options — Disable

Save/Load sounds: I:\~\SOUNDS\

Save TGA images: I:\~\IMAGES\

Create Folder with Anim name — Auto Name

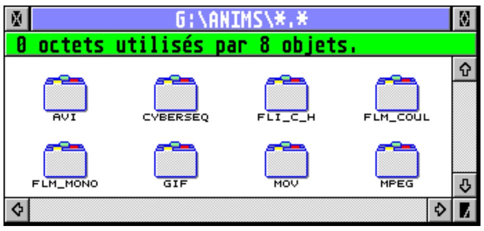
Converted Anims: I:\~\CONVERT\

Warnings · Set Sound · Save · Ok

Default path

You can set here the default path. When M_PLAYER is run, this is the first path that will be used in the fileselector.

In this example it points to my anim collection on drive G :


G:\ANIMS\*.*

0 octets utilisés par 8 objets.

AVI · CYBERSEQ · FLI_C_H · FLM_COUL · FLM_MONO · GIF · MOV · MPEG

Always count MPG frames

If your system is fast enough, you can set this option and frames will be counted when loading and MPEG video.

If it is not set, you let M_PLAYER decide, according to the power of your machine, the limit above witch it won't count automatically.

Sound system

M_PLAYER detects your sound hardware through the cookies.

_SND :	if bit 0 is set, then **Yamaha** available
	if bit 1 is set, then **DMA** available
PSND :	if present, a **Pcard** is available.

According to this, the different options appear enabled or disabled.

When loading M_PLAYER.OPT, if the saved value doesn't match the current hardware, or if no option file is present, then M_PLAYER selects for you the best hardware :

- ✔ DMA/SAGA is prefered *(no time consuming and good quality)*
- ✔ Psound is the second one *(an interrupt with fast routine, medium quality)*
- ✔ Yamaha is the last one *(an interrupt with large routine, low quality)*

If you change your sound setting, you'll be warned to reload the animation (if any is loaded in the main dialog).

You can ignore this if the animation has no sound or if you won't replay it (for example if you just disassemble the file in TGA images).

MJPEG frames per second

The MJPEG files don't include any timing information. On this line you can specify the default frame rate from 0,1 to 99,9.

Example:

0,1 frame per second means 10 seconds for each frame. This is a slideshow.

Max button (for MJpeg)

Again on slow machines, you may not want to respect a frame rate with MJPEG files, but instead watch all frames one by one. In this case, select this button that validates "Max speed" as default setting in the main dialog instead of "xx.x frames/sec".

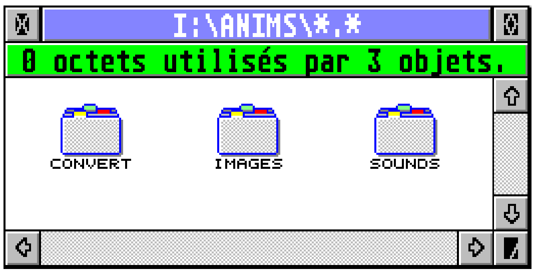
Enh button

When an Enhanced file is detected and valid, the Enhaced button will be selected *(if Enh is turned on)* or unselected *(if Enh is turned off)*. So this lets you define de default setting.

Win button

When replay in a window is available, the "W" button will be selected *(if Win is turned on)* or unselected *(if Win is turned off)*. So this lets you define de default setting for the very first time only. Then the setting of the "W" remains what you selected before.

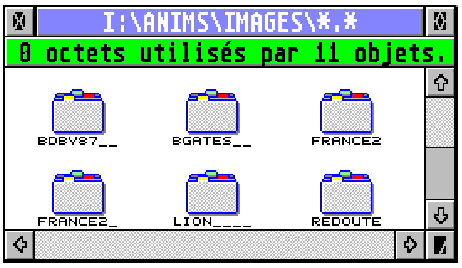
IX.2 Save and convert options



These are very important and will ease your work. Here is the organisation of my video working drive I :

I want every sound to be loaded or saved into the SOUNDS folder, every TGA saved into IMAGES and every conversion video saved into CONVERT.

These are the paths I have selected.



Create Folder with Anim name

If this is set, it applies to the TGA saving. A sub folder will be created with the anim name and there your images will be saved.

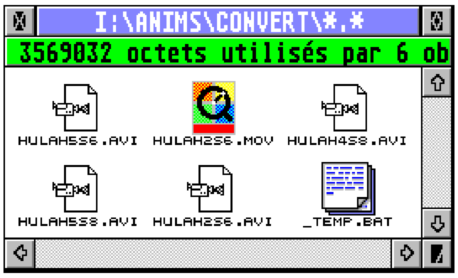
For example, when I disassembled BDBY87.AVI, the folder BDBY87__ was created and the images were saved inside it.

Auto Name

If this is set, then no fileselector is open and the names are computed automatically.

For a sound :
The name it the same as the video with AVR extension.

For TGA images :
The name takes the first characters of the video name and the last ones are the image indice. The number of digits is set according to the number of frames available.
If this number is unknown (large MPG), 7 digits are used.
For example, if `MYVIDEO.MOV` has 1520 frames, then the image will start at `MYVI0001.TGA`.



For Video conversions :
The last three characters and the extension summarize all the settings.
With `xxxxxQZB.EXT`
Q is the quality from 1 to 5.
Z is the zoom factor (H for half, D for double and S for same)
B is the number of bits (8 for 8 bits and 6 for 16 bits compression)
EXT will be either MOV or AVI.
For example : `HULA2S6.MOV` is a conversion from `HULAHOOP.MPG` with **[2]** quality=2, **[S]** same size and **[6]** 16 bits MOV.

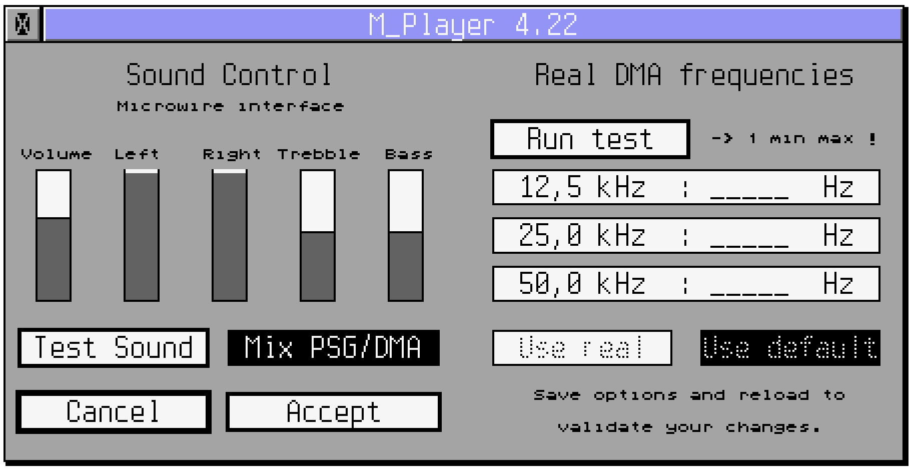
Disable

If this is set, all save and convert options are temporaly disabled but not deleted.
So you get the fileselectors and have to fill the names by yourself.

Warnings

If for you some warnings are annoying and that you know how to use M_PLAYER, you can disable this button. You won't get them anymore.

Set Sound



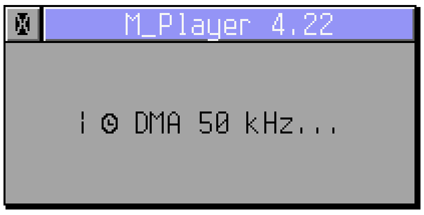
Save options and reload to validate your changes.

If you **own a TT**, then the Microwire Interface is available to control the sound and this button becomes available, it leads you to this dialog.

If you have a machine with the XBIOS routine "Soundcmd()" (such as **the Falcon**), the panel is limited to the left/right volume controls. The same for the SAGA of the **Apollo Vampire**.

To use "Test Sound", your application folder must contain the file M_PLAYER.AVR witch contains data for a 25 KHz, stereo sound, 8bits signed.

On the right side, you can **test the real DMA frequencies** of your machine. Compared to the original definition of the DMA STE sound, **newer machines** have a slightly **different tuning** that can lead to **unaccurate synchronization** between images and sound.

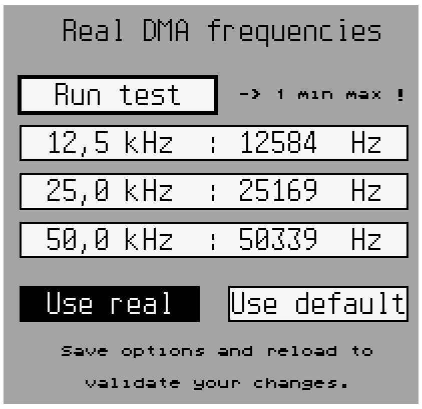


M_Player 4.22

DMA 50 kHz...

M_PLAYER includes the frequencies for the **TT**, the **Falcon** and **Aranym** *(limited to what I observed on my own computer)*. If you notice a sound lag, then clic on Run test.

Three passes will be performed, each one lasting 20 seconds or less, for the 50, 25 and 12.5 kHz sound.



Real DMA frequencies

Run test -> 1 min max !

12,5 kHz : 12584 Hz

25,0 kHz : 25169 Hz

50,0 kHz : 50339 Hz

Use real Use default

Save options and reload to validate your changes.

Then, the display will show you the results. Here you can see **what is expected on a TT**.

Notice that the sound is **a bit faster** than on a STE.

You can clic on Use real to use those values.

You can clic on Use default to keep what's built inside M_PLAYER.

Using Accept only sets the Microwire, if you want to use the new frequencies, you have to **save** the options **and reload** the program.

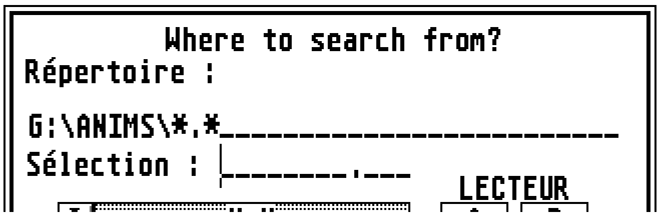
SAGA: For now, real frequencies calculations are not available on the **Apollo Vampire** as there is no way to know if a sound is stopped. Then, the program can't compute the duration of a sound.

Find videos on disk

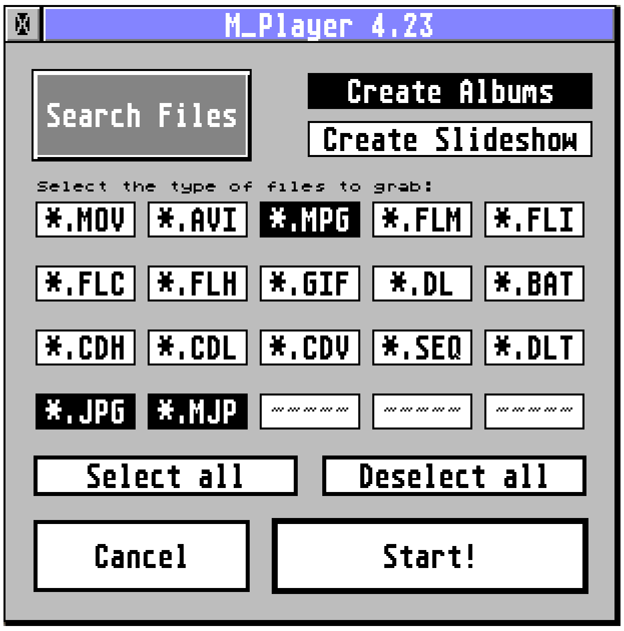
X Search a disk for animations

You can create a **batch file** or a set of **albums** that will contain every animation found into a disk or a folder.

The fileselector opens and ask you for the place where it should start to search. As an example, I select G:\ANIMS\ and validate my choice.



```
Where to search from?
Répertoire :

G:\ANIMS\*.*________________________
Sélection : ________.___
                                LECTEUR
```



```
M_Player 4.23

Search Files          [✓ Create Albums]
                      Create Slideshow

Select the type of files to grab:
*.MOV  *.AVI  [✓ *.MPG]  *.FLM  *.FLI
*.FLC  *.FLH  *.GIF  *.DL  *.BAT
*.CDH  *.CDL  *.CDV  *.SEQ  *.DLT
[✓ *.JPG]  [✓ *.MJP]  ~~~~~  ~~~~~  ~~~~~

Select all        Deselect all

Cancel            Start!
```

Note: *you can also start a search when you're in the fileselector for replaying videos. To do this, select the path you want in the fileselector, the filename field won't be used.* ***Then press the Alternate key and validate****. This was the old way to do it before the General Menu was created.*

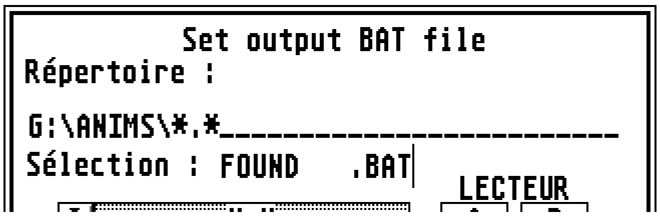
A dialog opens and lets you select the type of files you want to search for.
Here an example with MPG, JPG and MJP files wanted.

You have to decide wether you want **Albums** or a **Slideshow**.

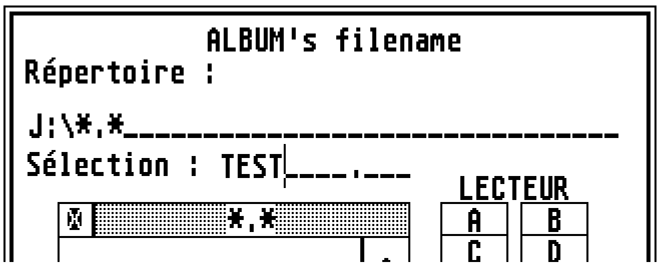
Then press **Start !**

If you selected **Slideshow**, you are asked for a name to save the batch file. Then the search begins.

This **includes all subdirectories** of the one selected.

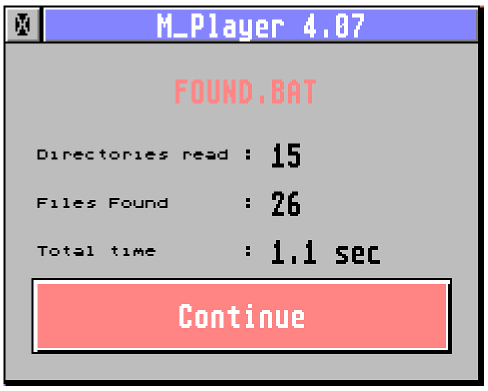


```
Set output BAT file
Répertoire :

G:\ANIMS\*.*________________________
Sélection : FOUND   .BAT
                                LECTEUR
```



```
ALBUM's filename
Répertoire :

J:\*.*______________________________
Sélection : TEST____.___
                                LECTEUR
            *.*                 A    B
                                C    D
```

If you selected **Albums**, you are asked the album's name. In this case the file extension will be ignored as it is provided by the program.
In the example here, the first album will be named TEST.000, the second TEST.001 and so on according to the number of files found.

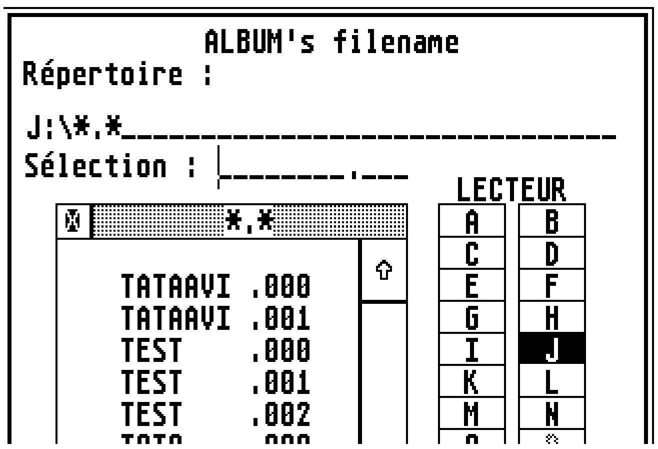


```
Directories read : 15
Files Found      : 26
Total time       : 1.1 sec
```

At the end, M_Player tells you how many directories were searched and how many files were found.

For a **slideshow**, you can validate and **automatically, the next fileselector points to your batch file** if you want to view all the anims that were found.

They are all replayed in sequence with a black background, no window.



```
Répertoire :
J:\*.*
Sélection :
TATAAVI .000
TATAAVI .001
TEST    .000
TEST    .001
TEST    .002
```

If you selected **Albums**, you can enter the corresponding section via the **General Menu** and load them. The example above leads you to three albums created:

TEST.000 to TEST.002.

More infos here: Albums

X.1 Slideshow of animations

A batch file for a slideshow of animations is a text file organized this way :

`SLIDEANI` on the very first line and column
`path+name of first animation`
`path+name of second animation`
...
`.stop` period stop on the last line.

The batch file can be edited and expanded with other features such as loops or numbered file names.

Each command must start with a period. Valid commands are:
`.rept xxx, .endr, .incr, .decr, .disp, .stop`

Example:

```
SLIDEANI
e:\anim\intro.avi
.rept 5
e:\anim\me.avi
e:\anim\you.avi
.endr
e:\anim\end.avi
.stop
```

This file will start with '`intro.avi`', then will display 5 times **`me.avi`** and **`you.avi`**, finally it will display **`end.avi`**.

<u>Numbered files example:</u>

```
SLIDEANI
e:\anim\intro.avi
e:\anim\scene000.avi
.rept 50
     .incr
     .disp
.endr
e:\anim\end.avi
.stop
```

This file will start with '**`intro.avi`**', then it will display 51 scenes named '**`scene000.avi`**' to '**`scene050.avi`**', and it will end with '**`end.avi`**'.

<u>IMPORTANT NOTE</u>

No resolution switch is made between two animations, no 'clear screen' either. So, you must be sure that all of your animations will be displayed into the same resolution and that every new one will cover the previous one to avoid garbage on the screen and crashes of the system.

XI EASY_BAT.PRG (discontinued)

EASY_BAT is a *semi-friendly* **tool to create BATCH files**.
The sequence of dialog boxes can't be altered, so if you miss something, then you have to do it all again. That's why it is semi-friendly…

The program is discontinued but still available. You should prefer the integrated panel or directly modify the batch file for specific features.

<u>Example :</u> let's go back to the VR-movie creation with the 18 views of the dice as seen in QuickTime VR movies. Let's run EASY_BAT, after the copyright screen, you get this :

XI.1 Image type selection

Here I select **TGA** to create a MOV with RLE16 colors.

Only MOV files can handle the VR extension.

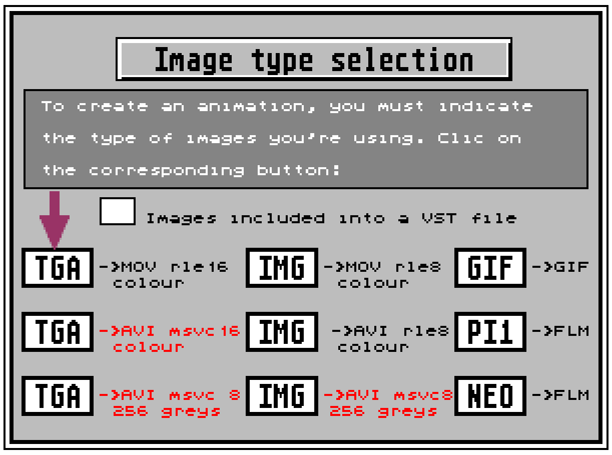


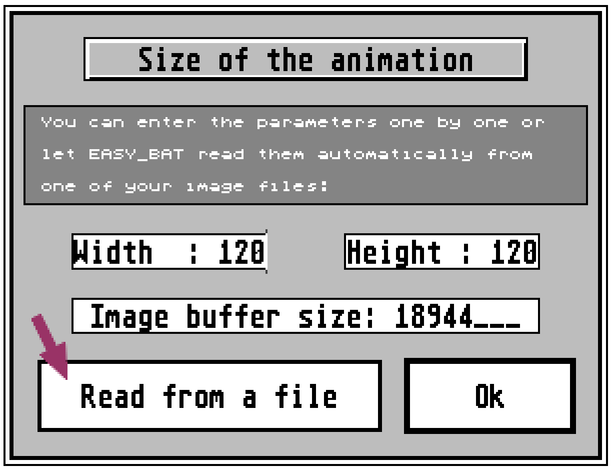
Image type selection

To create an animation, you must indicate the type of images you're using. Clic on the corresponding button:

XI.2 Size of the animation

Here I clic in **Read from a file** and in the fileselector, I enter the folder containing my images and just write *.TGA.

It fills automatically the three fields.



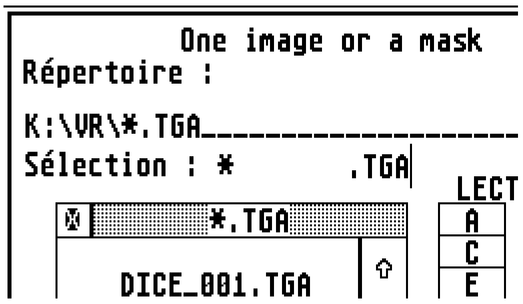
Size of the animation

You can enter the parameters one by one or let EASY_BAT read them automatically from one of your image files:

Width : 120

Height : 120

Image buffer size: 18944___



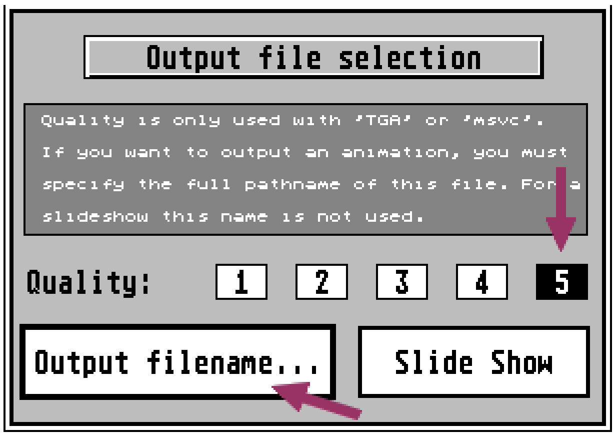
One image or a mask

Répertoire :

K:\VR\*.TGA____________________

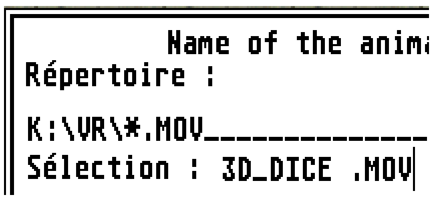
Sélection : * .TGA

XI.3 Output file selection


Output file selection

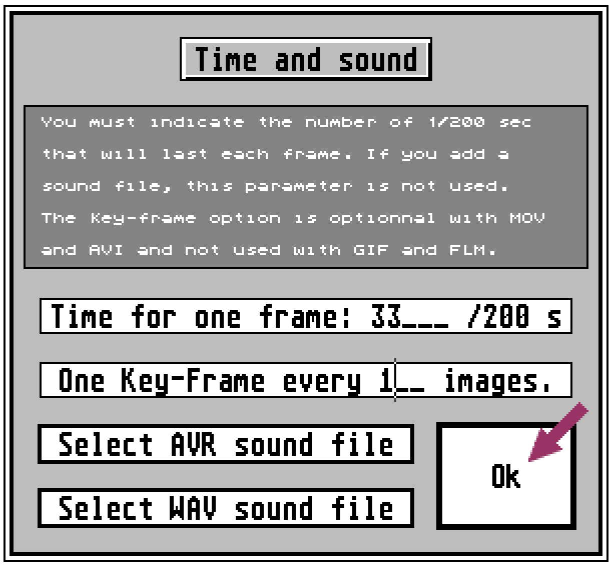
Quality is only used with 'TGA' or 'msvc'. If you want to output an animation, you must specify the full pathname of this file. For a slideshow this name is not used.

Here I select **quality 5** *(the best)* and then I clic on **Output filename** and give a name to the movie that will be created.


Name of the anim
Répertoire :
K:\VR\*.MOV______________
Sélection : 3D_DICE .MOV

If I just want a slideshow of images, I can ignore everything and clic on **Slide Show.** In this case, the BAT file won't allow you to create an animation, just to view the images in sequence.

XI.4 Time and sound


Time and sound

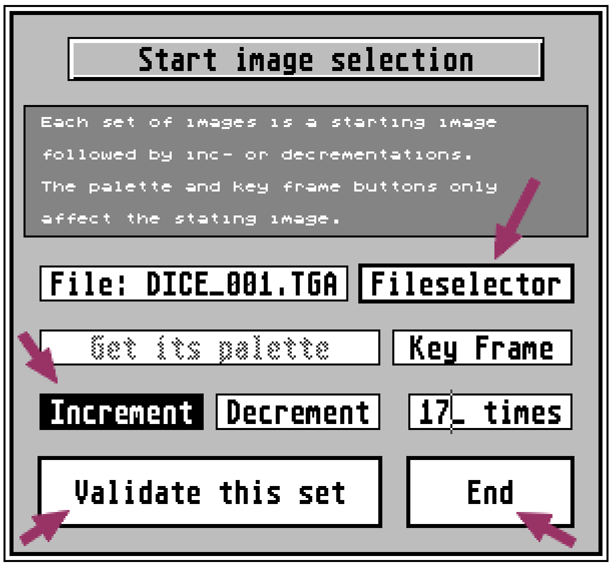
You must indicate the number of 1/200 sec that will last each frame. If you add a sound file, this parameter is not used. The Key-frame option is optionnal with MOV and AVI and not used with GIF and FLM.

Time for one frame: 33___ /200 s
One Key-Frame every 1__ images.

If an animation contains a sound, then the frame rate is automatically computed to fit the sound duration.

Here, no sound, so I write 33/200 sec for one image (6 fps) and I put **every frame as a key frame.**

XI.5 Start image selection


Start image selection

Each set of images is a starting image followed by inc- or decrementations. The palette and key frame buttons only affect the stating image.

File: DICE_001.TGA
17_ times

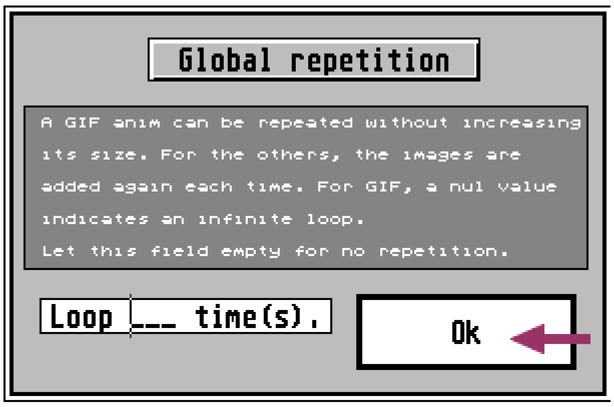
In this case, the images have all contiguous numbered names. So I just have to select the first one with **Fileselector**.

Then I indicate that this name will be **incremented 17 times** (up to **DICE_018.TGA**).

I **validate this set**.

Then **End**, no more images required.

XI.6 Global repetition



Global repetition

A GIF anim can be repeated without increasing its size. For the others, the images are added again each time. For GIF, a nul value indicates an infinite loop. Let this field empty for no repetition.

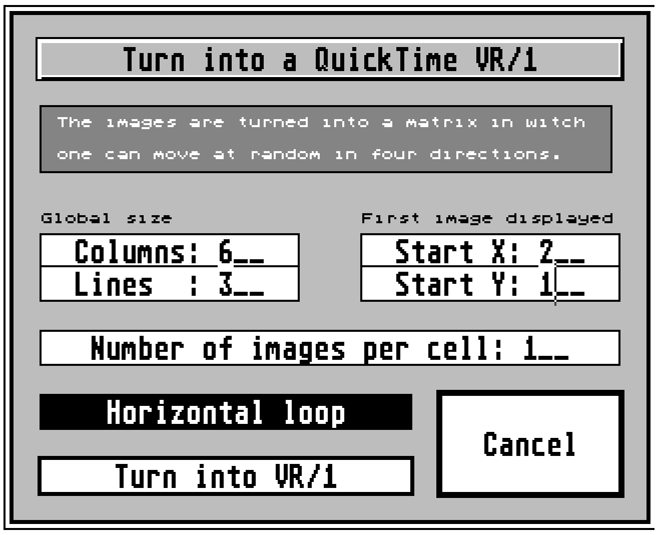
Loop ___ time(s).

Ok

In this case, it is not appliable.
So I let a blank field and Ok.

This is mostly used with GIF movies.

XI.7 Turn into a QuickTime VR/1



Turn into a QuickTime VR/1

The images are turned into a matrix in witch one can move at random in four directions.

Global size

Columns: 6__
Lines : 3__

First image displayed

Start X: 2__
Start Y: 1__

Number of images per cell: 1__

Horizontal loop

Turn into VR/1

Cancel

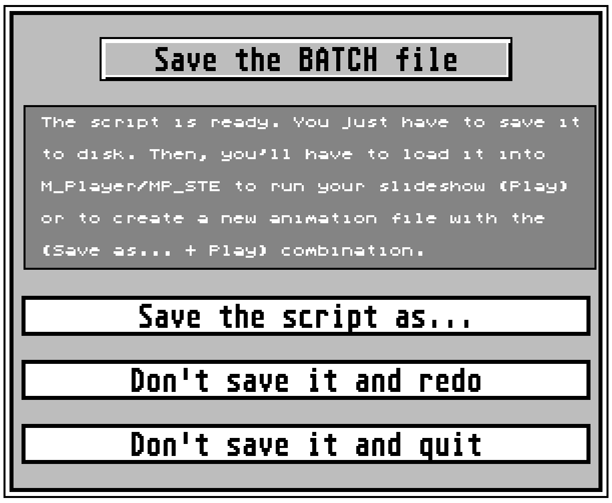
The special box for **VR informations**.
I set the **number of columns and lines**.
I set the **initial image coordinates.**

There is **only 1 image per cell,** this means that they are fixed views.

I enable the **horizontal loop.**

Finally, I clic on **Turn into VR/1**.

XI.8 Save the batch file



Save the BATCH file

The script is ready. You just have to save it to disk. Then, you'll have to load it into M_Player/MP_STE to run your slideshow (Play) or to create a new animation file with the (Save as... + Play) combination.

Save the script as...

Don't save it and redo

Don't save it and quit

There we are, the BAT file is ready.

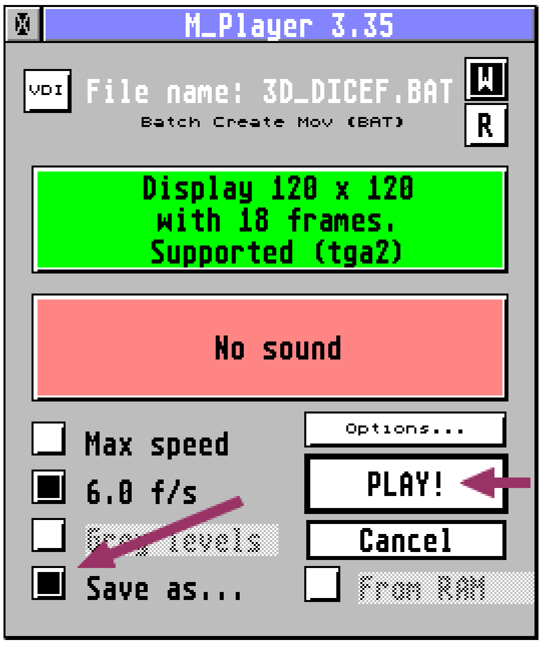
I clic on **save**, and I give it the name 3d_DICEF.BAT (F for fixed images).

XI.9 Create the animation

You have to load the BAT file in M_PLAYER.

Ensure that you have checked the « **Save As...** » button to **enable the creation**.

Clic on **Play !** The creation starts



M_Player 3.35

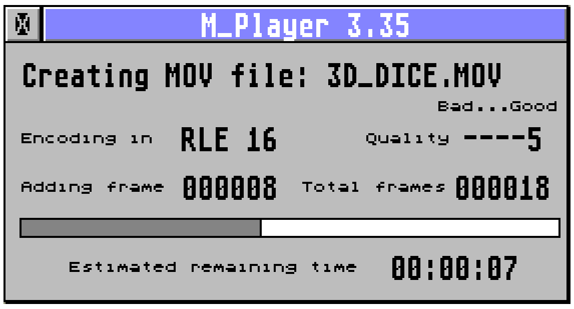
VDI File name: 3D_DICEF.BAT W
Batch Create Mov (BAT) R

Display 120 x 120
with 18 frames.
Supported (tga2)

No sound

☐ Max speed
[✓ 6.0 f/s]
☐ Grey levels
[✓ Save as...]

Options...
PLAY!
Cancel
☐ From RAM



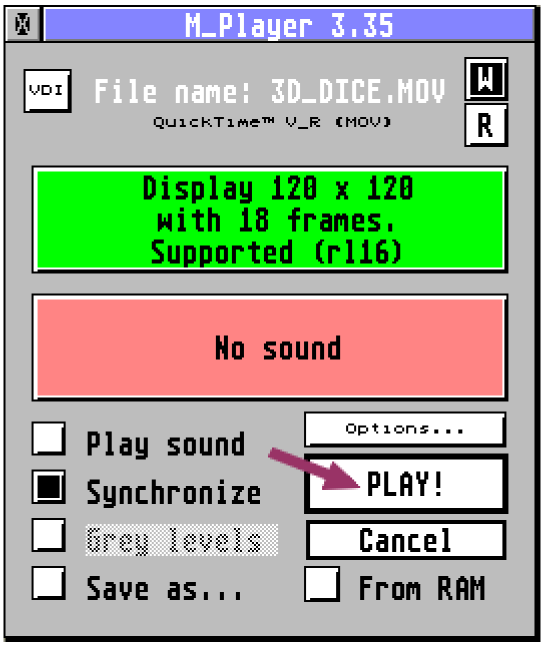
M_Player 3.35

Creating MOV file: 3D_DICE.MOV
Bad...Good
Encoding in RLE 16 Quality ----5
Adding frame 000008 Total frames 000018

Estimated remaining time 00:00:07

XI.10 Replay the animation

Load the 3D_DICE.MOV animation in M_PLAYER, notice that under the name the QuickTime VR format is detected.



M_Player 3.35

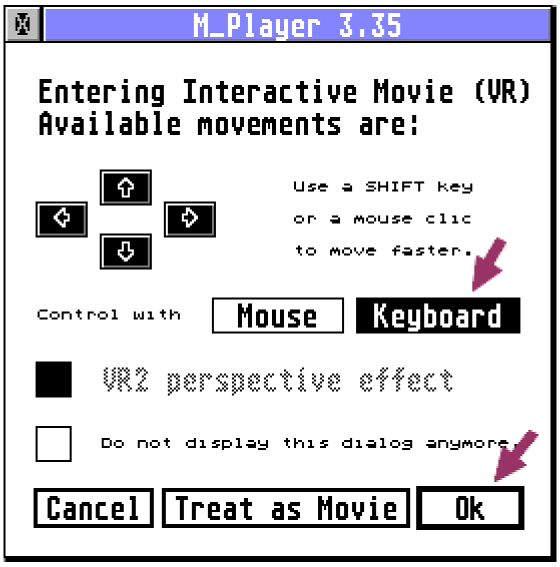
VDI File name: 3D_DICE.MOV W
QuickTime™ V_R (MOV) R

Display 120 x 120
with 18 frames.
Supported (rl16)

No sound

☐ Play sound
[✓ Synchronize]
☐ Grey levels
☐ Save as...

Options...
PLAY!
Cancel
☐ From RAM



M_Player 3.35

Entering Interactive Movie (VR)
Available movements are:

Use a SHIFT key
or a mouse clic
to move faster.

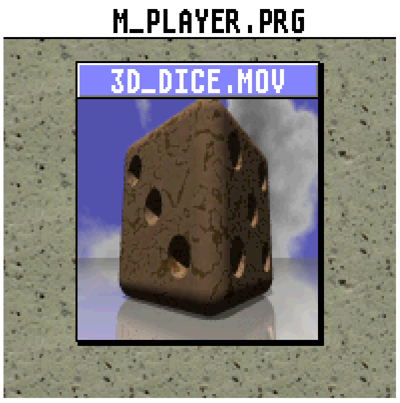
Control with Mouse [✓ Keyboard]

[✓ VR2 perspective effect]

☐ Do not display this dialog anymore

Cancel Treat as Movie Ok

I select **Keyboard** to move the dice, then **Ok**.



Finally, the animation appears and I can **move around** the dice left, right, up and down.

XII Command line options

The command line is:

```
<options> path\file_name
```

The options must appear before the file name! Options can be separated with spaces or not.

Every option is:

```
+LETTER to set this option
-LETTER to clear this option
```

Available options:

+d, -d Display Dialogs

if set will display the dialogs, if cleared won't display them.
Default value: +d

+p, -p Play Sound

if '-d' is specified, then if set will play the sound (if available), if cleared won't play the sound
Default value: +p

+s, -s Synchronize

if '-d' is specified, then if set will synchronise the images and sound, if cleared will display the images at the max speed.
Default value: +s

+a, -a Save As...

if '-d' is specified, then set « Save As... » to enter step by step mode automatically or to create a movie with a batch file.
Default value: -a

+e, -e Error Messages

if '-d' is specified, then if set will display every error message (bad file, etc...), else if cleared, won't display any error message.
Default value: -e

+i, -i Repeat

if '-d' is specified, then if set will play in loop mode (on the mask or single file passed through the command line) until an event accurs (keyboard, mouse, joystick), else will play normally.

Default value: -i

This is for being used with a screen saver for example.

+xnnn, +ynnn Coordinates

if '-d' is specified, then fixes the position where to display the anim. If a resolution switch is necessary, then this position will be ignored and the anim will be centered as usual.

In this mode, the background is not cleared, so you can display an anim into one of your windows. There are no checking upon these coordinates, you must be sure that the anim fits into the screen.

Note: *when using a TT without graphic card, the TT Low mode is often used, and some anims are doubled in height to correct the proportions, make some tests!*

Note2: *+x100 or -x100 are equivalent, both x and y must be specified, else the coordinate is ignored.*

Note3: *when using a TT without graphic card, the X value is automatically aligned to a 16 pixels boundary.*

Options p, s, a, i and e are only usefull when '-d' is specified, else, the player uses the returned values from the dialogs to set the options.

+r nnnn Return infos

(+r and -r are equivalent)

specifies the address of a buffer where M_Player will return the infos about a movie, in this mode the anim is not played, you'll get no dialog, only the buffer filled (72 bytes are needed):

WORD status 1=Ok
0=File not found (nothing else filled)
-1:unknown type (only file_name filled)
CHAR file_name (14 bytes: max 12 + nul + even address)
CHAR file_type (28 bytes: max 26 + nul + even address)
such as 'Video for Windows (AVI)'
WORD graph_status
1: supported
0: no graphics (infos filled with garbage)
-1:unsupp (infos filled)
WORD width
WORD height
LONG number of frames
LONG compression (4 bytes such as 'cram', 'cvid'...)
WORD sound_status
1: supported
0: no sound (infos filled with garbage)
-1: unsupp (infos filled)
WORD sound bits (more often 8 or 16)
WORD channels (1:mono, 2:stereo)
LONG frequency

LONG version
4 ASCII bytes representing the version of the player.

Examples:

```
m_player -d *.AVI
```

will display every AVI from the current folder without any dialog and using the default values (+p, +s, -a).

```
m_player -d +a c:\batch\*.bat
```

will create MOV files from every BAT file found into the c:\batch folder.

```
m_player -d-p+s+i C:\anims\*.*
```

will display every file found into the anims folder without sound but at their normal speed (synchro is on). Will also loop into this folder until an event occurs (keyboard, mouse, joystick), it means that when the end of the folder is reached, the research starts again.

```
m_player +r81500 C:\anims\*.DL
```

will fill the buffer with the infos corresponding to the first DL anim found into C:\ANIM. If no DL file is found, the first word will contain -1 and garbage for the other bytes.

Note: for options using a number as parameter (+x, +y, +r...), you can write:

`+x200`	decimal value
`+x$C8`	hexadecimal value (with $)
`+xw??`	w indicates that the two bytes ?? represent a WORD value
`+xl????`	l indicates that the 4 bytes ???? represent a LONG value

The **'w' and 'l' facilities** allow you to pass parameters without the Integer to string conversion, just 'poke' them into the command line.

XIII Calling M_PLAYER as an accessory

When **M_Player is an accessory**, it waits for two messages:

AC_OPEN (40)
when called from the menu bar of the desktop

M_PLAYER ('MP' or $4D50)
when called by another application, the AES buffer must be like this:

word(0)	: $4D50
word(1)	: appl_id of the calling process
word(2)	: 0 (not more than 16 bytes)
word(3) and	
word(4)	: LONG address of the command line
word(5)	: return message or zero.
word(6)	; not used
word(7)	: not used

If the **LONG address** is 0, then no command line is used and you'll get the normal dialogs and fileselector. If an address is specified, it should point to a nul-terminated string corresponding to the command line as described above.

For example: `"-d+x100+y100 d:\anim\test.flm"`

to display the anim TEST.FLM at (100,100) without clearing the background and without displaying the dialogs.

return_message, if set to 0, is unused. But, if set to anything else will cause M_PLAYER to send back to your application the corresponding message when the replay is finished.

For example, if return_message='ND', your application will recieve the message 'ND' (with evnt_mesag) when M_Player has finished.

XIV Supported file formats

Here is the list of what M_PLAYER can read.

QuickTime *.MOV (VR1 Objects & Vr2 Panoramas supported)

Video codecs : CVID, RLE1, RLE2 (grey and color), RLE4 (grey and color),
RLE8 (grey and color), RLE16, RLE24, RLE32, SMC8 (grey and color)
RAW1, RAW2 (grey and color), RAW4 (grey and color),
RAW8 (grey and color), RAW16, RAW24, RAW32, RPZA (15 bits)
WRLE (256 colors), MSVC8 (grey and color), MSVC16
YUV2, YUV9, YVU9, **JPEG, MJPA, MJPB**
*(For **JPEG, MJPA** and **MJPB**, the external module JPEG_68K.RIM must be located in the application folder).*
Audio codecs : TWOS, RAW (8/16 bits, mono/stereo), IMA4

Video for Windows *.AVI

Video codecs :CVID, CRAM16, CRAM8, MSVC16, MSVC8
RLE8, RGB8, YUV9, YVU9, IV32,**JPEG, MJPG**
*(For **JPEG, MJPG**, the external module JPEG_68K.RIM must be located in the application folder).*
Audio codecs :TWOS, RAW (8/16 bits, mono/stereo)

JPEG *.JPG

Unique images. (*JPEG_68K.RIM required)*

MJPEG *.MJP

Video codec : JPEG (*JPEG_68K.RIM required), sequences of concatenated images.*

MPEG *.MPG

Video codec : mpeg1, mpeg2
No support for audio.

Extended Lexicor Film *.FLM

Video codecs : ST Low, ST High, TT Med
Audio codecs: DMA 8 bits mono/stereo, 12,5kHz, 25kHz, 50kHz.

Compuserve *.GIF

Video codecs : GIF87, GIF89

Video Master *.FLM

Video codecs : VMAS, VFAL
Audio codecs : 8 bits mono, any frequency

CD Stratos Magazine *.CDH, *.CDL, *.CDV

Video codecs : ST Low, ST High, Falcon 256

Audio codecs : DMA sound 8 bits mono 12,5kHz

Old ST Low Atari animations

➢ *.SEQ	(Cyber Paint)
➢ *.PI1 + *.DLT	(Cyber)
➢ *.FLM	(Kinetic Microsystems)
➢ *.FLM	(Lexicor)

Old PC animations

➢ *.FLI/FLC/FLH	(Autodesk Animator)
➢ *.DL	(types 1 and 2)

Batch Files *.BAT

For slide shows or creation of animations.

Images : TGA2, TGA10, 16 or 24 bits, uncompressed or RLE
GemXIMG 256c, GIF, PI1, NEO

Sound : AVR or WAV, uncompressed.

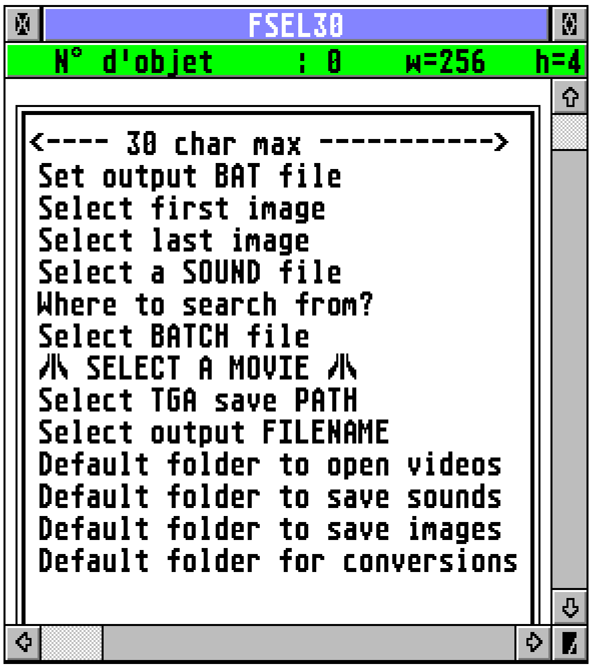
XV Translating the interface

As of version 4.09, every single text has been removed from the executable file to be stored into the RSC file. This way, one can easely translate the whole interface including the alert messages.

XV.1 File selector titles

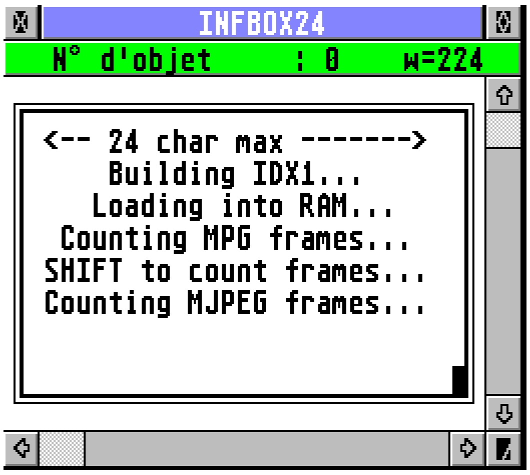
The fileselector can display a title with a maximum of 30 characters to guide the user.

You'll find all those titles into the FSEL30 tree. A first line remembers you the limit you can't bypass.



```
FSEL30
N° d'objet      : 0     w=256    h=4

<---- 30 char max ----------->
 Set output BAT file
 Select first image
 Select last image
 Select a SOUND file
 Where to search from?
 Select BATCH file
   SELECT A MOVIE
 Select TGA save PATH
 Select output FILENAME
 Default folder to open videos
 Default folder to save sounds
 Default folder to save images
 Default folder for conversions
```

XV.2 General info box

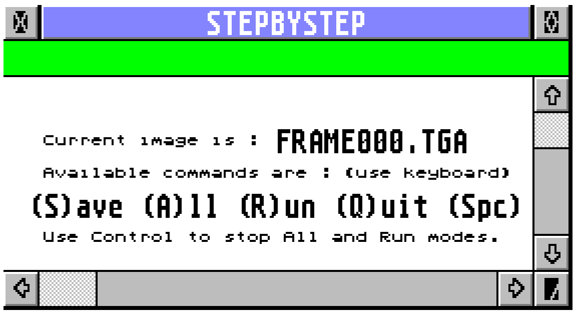


```
INFBOX24
N° d'objet      : 0     w=224

<-- 24 char max ------->
    Building IDX1...
   Loading into RAM...
 Counting MPG frames...
SHIFT to count frames...
Counting MJPEG frames...
```

One general information box is used and is currently limited to 24 characters.

If you can't fit your text into this limit, you must modify the corresponding tree GENINFO.

XV.3 Step by step box



```
STEPBYSTEP

Current image is : FRAME000.TGA
Available commands are : (use keyboard)
(S)ave (A)ll (R)un (Q)uit (Spc)
Use Control to stop All and Run modes.
```

This special box reminds you the keys you can use when disassembling a video. They appear enclosed in parenthesis.

This string is parsed by M_PLAYER to adapt the key to every language.

Be sure not to use twice the same key! In english, keys S, A, R and Q are used.

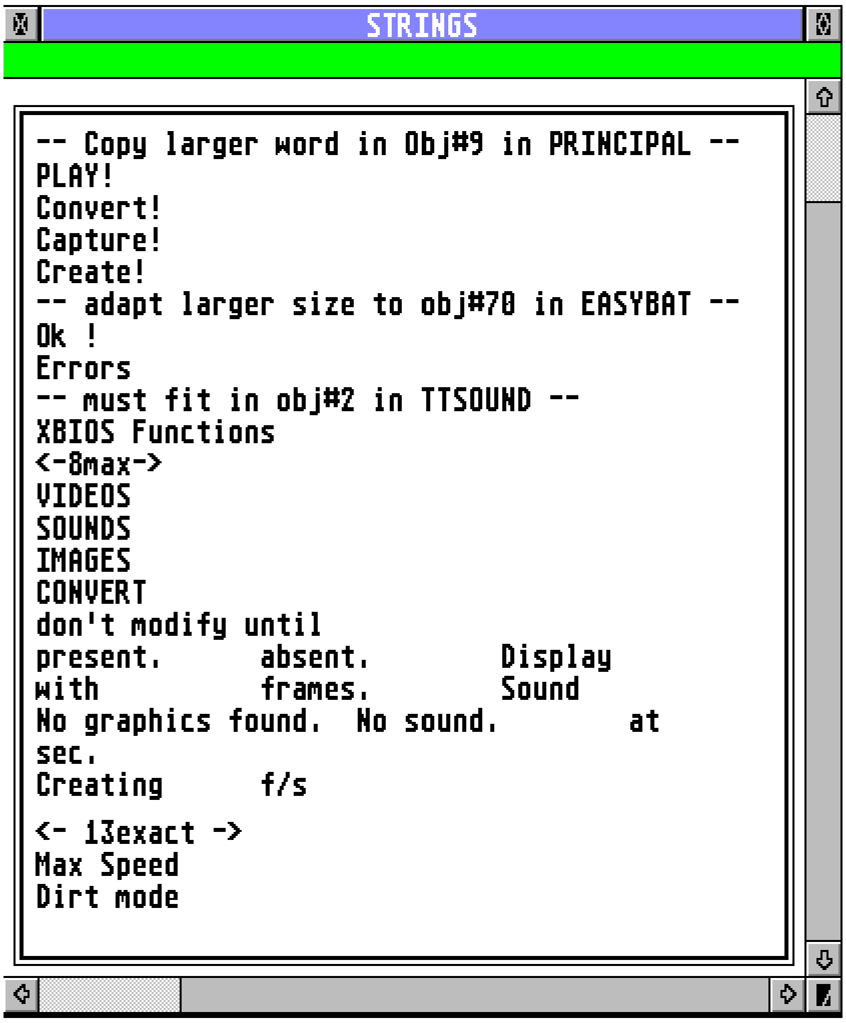
XV.4 Free strings

Lots of other strings are grouped into the STRING tree.

Some are just informations to guide you (telling you to copy one string into a specific tree or to limit your text to a certain number of characters).

Respect the strings that end with a period "." and those who don't.

Respect the uppercase letters (start of a sentence) and the lower case.

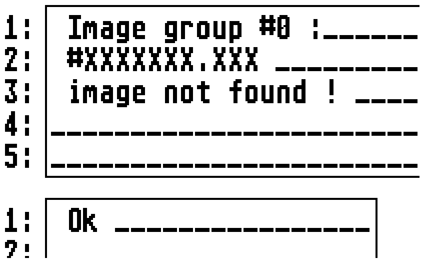


```
-- Copy larger word in Obj#9 in PRINCIPAL --
PLAY!
Convert!
Capture!
Create!
-- adapt larger size to obj#70 in EASYBAT --
Ok !
Errors
-- must fit in obj#2 in TTSOUND --
XBIOS Functions
<-8max->
VIDEOS
SOUNDS
IMAGES
CONVERT
don't modify until
present.      absent.       Display
with          frames.       Sound
No graphics found.  No sound.       at
sec.
Creating      f/s

<- 13exact ->
Max Speed
Dirt mode
```

XV.5 Alert strings

When translating alerts, you'll find some strange fields starting with "#". You must leave them as they appear including the spaces before and after if present.

M_PLAYER will look for those fields to display specific informations according to the context.

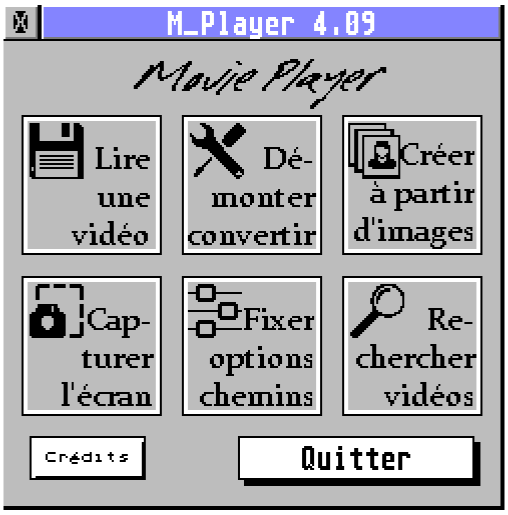


```
1: Image group #0 :______
2: #XXXXXXX.XXX _________
3: image not found ! ____
4: ______________________
5: ______________________

1: Ok _________________
2:
```

XV.6 General menu

For the general menu, there is no other way than redrawing the icons to change the text..!

I can help.


M_Player 4.09
Movie Player
Lire une vidéo
Dé-monter convertir
Créer à partir d'images
Cap-turer l'écran
Fixer options chemins
Re-chercher vidéos
Crédits
Quitter